UNIVERSIDAD AUTÓNOMA DE MADRID

ESCUELA POLITÉCNICA SUPERIOR

Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación (EUR-ACE R)

TRABAJO FIN DE GRADO

Estudio de modelado estadístico de tráfico en Internet mediante el uso de distribuciones de Burr

Autor: Gonzalo López Segovia

Tutor: Luis de Pedro Sánchez

Ponente: Jorge Enrique López de Vergara

junio de 2021

Resumen

El continuo crecimiento del tráfico de Internet en las últimas décadas y el aumento masivo de intercambio de datos ha generado la necesidad de conocer su comportamiento y evolución. Esto ha conducido a una reorientación del análisis del tráfico de Internet hacia el monitoreo del tráfico y la prevención de ataques de hacking, como pudiera ser un ataque de denegación de servicio (DoS), tal y como ocurrió el 19 de marzo de 2013, cuando la organización *Cyberbunker* inundó de correos no deseados la firma de seguridad suiza *Spamhaus* y bloqueó el nodo central de Londres.

Para llevar a cabo esta tarea preventiva, es necesario poder analizar y caracterizar el tráfico de Internet y, en este sentido, el reconocimiento de patrones es una base fundamental para el análisis y prevención de ataques y anomalías registradas en este tráfico.

Este trabajo de investigación afronta el análisis de modelos estadístico de distribuciones Burr (de tres parámetros) para modelar series temporales referidas al tráfico de Internet capturado tanto en una red doméstica como en una red de un campus universitario. La distribución Burr ya ha sido estudiada en los campos de las finanzas y de los seguros, y aquí vamos explorar sus capacidades en el terreno del tráfico de red. En particular, nos proponemos demostrar que el tráfico de red se puede ajustar mediante una distribución Burr y que esta función de distribución es más eficiente en el cálculo que alfa-estable, otra distribución orientada a este mismo problema. Así, la distribución Burr podría servir para futuros trabajos dedicados a la monitorización y análisis de tráfico de red.

Para comprobar la calidad del ajuste, se realiza un remuestreo de la función ajustada mediante el algoritmo de *bootstrap*. Además, también se analiza el uso de mixturas multimodales de tipo Burr mediante el algoritmo de clasificación de datos de *K-Means*.

Por último, se hace mención de la comparación entre el comportamiento detectado en una red doméstica con respecto a una red de un campus universitario, sirviendo esto último como posible orientación para trabajos futuros centrados en el análisis de redes empresariales y la caracterización comportamientos anómalos y prevención riesgos.

Palabras clave: MATLAB, Wireshark, tshark, Burr, alfa-estable, k-means, tráfico de Internet, bootstrap, modelo estadístico, mixturas, bimodal, ping, distribución de probabilidad.

Abstract

The ongoing growth of the Internet traffic since the last few decades and the massive increasing of data exchange has generated the need of knowing its behaviour and development. Besides, Internet traffic analysis must be also aimed at monitoring the Internet traffic as well as the prevention of hacking attacks such as a DoS (*denial of service*) as it happened on March 19th 2013 when a organization called *Cyberbunker* spammed the swiss security company *Spamhaus* and crashed the main node in London.

For this purpose, it is necessary to be able to analize and characterize Internet traffic. Pattern recognition is a fundamental basis in the analysis and prevention of attacks as well as to prevent other abnormalities occured in Internet traffic.

This research study takes on the study of the analysis of statistical models based on Burr Distribution XII (from 3-parameters family) to model time-series referred to Internet traffic captured both in a house network (LAN) and in a university network. Burr distribution has been studied in fields of finances and insurance studies, and we intend to show its capabilities in the field of network traffic.

The main goal of this research is to prove that the network traffic can be fitted by a Burr distribution and this distribution function is more efficient thant alpha-estable, another distribution used to model Internet traffic. For this reason, it could be useful for future researches dedicated to the monitoring and analysis of network traffic. The quality of the fitting is tested using the *bootstrap* algorithm. Moreover, there is a part dedicated to the research and analysis of Burr mixtures with the *K-Means* clustering algorithm.

Finally, it is mentioned the comparative between the behaviour detected in a LAN network versus a University network. This could be useful for futures research focused on the analysis of corporate networks and the characterization of abnormal behaviours and risk prevention.

Key words: MATLAB, Wireshark, tshark, Burr, alpha-stable, k-means, Internet traffic, bootstrap, statistical model, Burr mixtures, bimodal, ping, probability distribution.

Índice

Índice de figuras

1 Introducción

1.1 Motivación

En marzo de 2013, la empresa holandesa *Cyberbunker* lanzó un ataque masivo de denegación de servicio contra *Spamhaus*, la firma suiza de seguridad, inundando la red de correos basura, provocando una ralentización general de Internet y llegando a afectar a puntos clave como el nodo central de Londres[1].

Este hecho ilustra cómo el monitoreo del tráfico en redes empresariales es un aspecto clave para la prevención de ciberataques como pueden ser los de denegación de servicio[2], cada vez más habituales en este tipo de redes. Un ataque, como el citado en el párrafo anterior, a unos nodos estratégicos puede llevar a consecuencias catastróficas si no se consigue detener. Para poder prevenir un ciberataque de este estilo, se debe primero conocer y caracterizar el comportamiento genérico que tiene una red de computadoras, bien sea mediante el análisis sobre una red local doméstica o sobre una red empresarial.

Se ha demostrado que la función alfa-estable modela y ajusta de manera fidedigna el tráfico en Internet. Sin embargo, se trata de una función de distribución de gran coste computacional, poco eficiente y lenta. Por este motivo, se ha decidido explorar otros métodos de ajuste del tráfico en Internet como es el caso de la distribución Burr.

Este trabajo pretende mostrar que el tráfico de Internet cuando sigue una distribución de cola pesada puede caracterizarse con una función de distribución Burr.

1.2 Objetivos

El principal objetivo de este trabajo consiste en analizar el comportamiento del tráfico de red tanto en una red LAN como en una de un campus universitario, y demostrar que se puede ajustar el tráfico de red mediante una distribución Burr.

También se trata de probar que el ajuste a la distribución Burr da buenos resultados y que puede servir como sustituto del ajuste en términos de la distribución alfa-estable para modelar el tráfico de Internet de una manera eficiente.

Por otro lado, se pretende mostrar cómo el tráfico de Internet puede llegar a ser caracterizado con un modelo de mixturas Burr y ser aplicado en el estudio de diferentes comportamientos en el tráfico de Internet.

1.3 Requisitos

Las principales tareas a desempeñar en este trabajo se refieren a la captura del tráfico de red, la posterior creación de una base de datos propia a partir de esas capturas de tráfico y el procesado estadístico de esos datos.

Para poder llevar a cabo esta labor, se ha requerido disponer de un entorno de trabajo orientado a este fin, poder conectarse a un ordenador de la red del campus universitario y adquirir conocimientos sobre GNU/Linux.

En particular, han sido fundamentales algunos conocimientos sobre:

- El funcionamiento del comando *ping*.

- El manejo y funcionamiento de *Wireshark*, un software analizador de tráfico de red, así como de su versión en ventana de comandos GNU, *tshark*, para la captura de los datos.
- El uso de *MATLAB*, un IDE orientado al procesamiento y análisis de datos como que trabaja sobre un lenguaje de programación interpretado con un potente motor gráfico.
- Conocimientos de análisis estadístico.

1.4 Plan de trabajo

	01/19	02/19	03/19	04/19	05/19	06/19	03/20	04/20	05/20	06/20
Captura de datos	x		x							
Código ajuste		x	x	x	x					
Código K-Means					x	x	x	x		
Análisis de datos				x	x	x	x	x	x	x
Documentación	x	x							x	x
Escritura									x	x

1.5 Estructura del documento

Esta memoria se organiza en los siguientes capítulos:

1. **Introducción.** Capítulo en el que se definen las motivaciones, los objetivos, los requisitos y la planificación del trabajo.
2. **Estado del arte.** Dedicado a resumir el estado del arte de las técnicas y conceptos matemáticos relacionado con el trabajo.
3. **Diseño y desarrollo.** Centrado en el planteamiento y diseño del problema, las diferentes aplicaciones utilizadas y los contratiempos encontrados durante el desarrollo del trabajo.
4. **Resultados y validación.** Dedicado al análisis y comentario de los resultados obtenidos para distintos casos de estudio.
5. **Conclusiones y trabajo futuro.**

2 Estado del arte y técnicas

En este capítulo se expondrán de manera resumida las herramientas básicas necesarias para la captura de tráfico de red así como algunas nociones teóricas del análisis estadístico relativas a las distribuciones alfa-estable y Burr. También se discutirán brevemente los algoritmos que se emplearán para clasificar los datos y el estado del arte en el campo de ajuste estadístico del tráfico de red en Internet.

El capítulo se divide en una sección dedicada a las técnicas de captura de tráfico de red con especial atención al comando ping, otra sección dedicada a explicar en qué consiste la distribución Burr y una última sección dedicada al algoritmo *K-Means*.

2.1 Técnicas de captura de tráfico de red: el comando ping

Para poder llevar a cabo este estudio, en primer lugar se debía conseguir una captura de datos de tráfico de red. Se tomó la decisión de que la variable que se iba a tratar para estudiar el comportamiento de la red fuesen los tiempos de llegada de los paquetes de respuesta ICMP, o mensajes *ping*. Este comando permite, mediante el envío constante de paquetes ICMP (Internet Control Message Protocol) cada segundo, la monitorización de los tiempos de llegada entre paquetes ICMP y, con ello, el estudio de la latencia de la red. El comportamiento del comando *ping* se ilustra en la figura 1[3]:

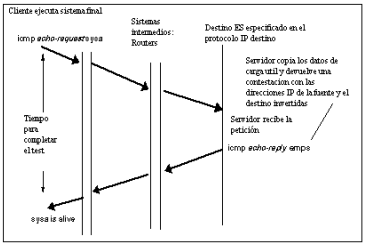


Figura 1: Comando `ping`

La Internet Engineering Task Force (IETF) define el Protocolo de Mensajes de Control de Internet ICMP en el documento RFC-792 [4]. Este protocolo se utiliza en el comando *ping* cuya finalidad es determinar si hay conexión entre el *host* y los equipos remotos (en este caso, servidores DNS)[5].

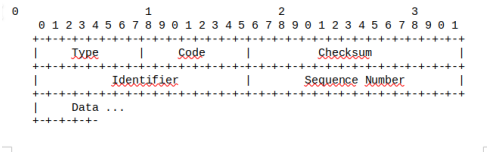
De acuerdo con la IETF, el protocolo IP contiene encapsulado el paquete ICMP [6] tal y como se muestra en la figura 2 de acuerdo con el estándar del datagrama de IPv4.

IPv4 Datagram

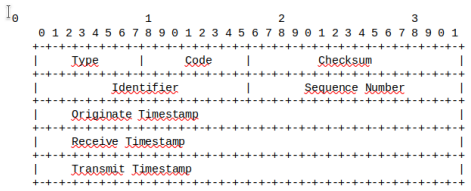
	Bits 0-7	Bits 8-15	Bits 16-23	Bits 24-31
Header (20 bytes)	Version/IHL	Type of service	Length	
	Identification		*flags* and *offset*	
	Time To Live (TTL)	Protocol	Header Checksum	
	Source IP address			
	Destination IP address			
ICMP Header (8 bytes)	Type of message	Code	Checksum	
	Header Data			
ICMP Payload (*optional*)	Payload Data			

Figura 2: Protocolo IP

La cabecera IP consta de 20 Bytes y la carga de datos (o *payload*), de 8. Es esta última la que se corresponde con el paquete ICMP encapsulado. Dentro del paquete ICMP, resultan de especial interés los campos reservados para el tipo de mensaje y de código que ocupan 2 Bytes para así poder filtrar y elegir los datos de tiempo de llegada de los paquetes de respuesta. A continuación se muestra el esquema que proporciona sobre los mensajes *Echo Reply Message*[4].



Un byte para el tipo de mensaje, otro para el código, dos para el checksum, dos para el identificador y dos para la secuencia numérica. Y el resto de bytes, reservado para los datos (longitud variable). El tipo de mensaje indica si se trata de un paquete *ICMP echo request* (tipo de mensaje = 8) o *ICMP echo reply* (tipo de mensaje = 0). Y los paquetes con mensajes de marca de tiempos se definen de la siguiente manera[4].



2.2 Técnicas estadísticas: distribuciones alfa-estable y Burr

En otros trabajos, como por ejemplo el realizado por Daniel Perdices[7], se ha estudiado el comportamiento del tráfico de red como una distribución de cola pesada. Una de esas distribuciones de cola pesada es la función alfa-estable cuya PDF y CDF vienen representadas en la figura 3.

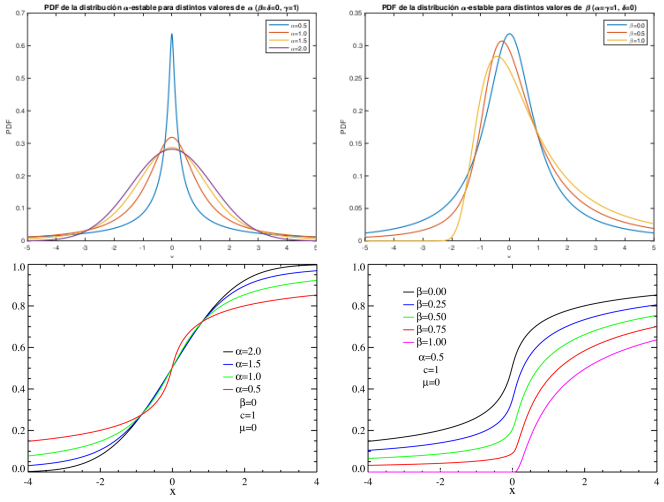


Figura 3: Distribución alfa-estable: superior: PDF, inferior: CDF; izquierda: dependencia con α, derecha: dependencia con β

Alfa-estable se caracteriza por cumplir que la combinación lineal de dos variables aleatorias con dicha distribución sigue también esta distribución[7]. Esta distribución resulta de gran utilidad porque permite modelar comportamientos muy variados, pero tiene el inconveniente de no contar con definiciones analíticas de sus PDF y CDF para todos los valores de los parámetros[8]. El hecho de que ambas propiedades se definan en términos de una integral de otra función para la que no se conoce su primitiva se traduce en un mayor coste de cálculo para su evaluación. Para solventar este problema, se plantea como alternativa la función de distribución Burr.

La distribución Burr Tipo XII[9], que es su nomenclatura registrada, pertenece a la familia de distribuciones de 3 parámetros: (α, c, k). A diferencia de otras distribuciones, en esta se desconoce el significado estadístico de cada uno de sus parámetros. Es una distribución que se usa también en los campos de las finanzas, en el análisis de seguros [10, 11], y en la hidrología[12].

La estimación de los parámetros está definida por defecto, según MATLAB, para un intervalo de confianza del 95%[9].

La función de densidad de probabilidad *pdf, probability density function* de la distribución Burr se define como:

$$f^{Burr}(x) = \frac{k\,c}{\alpha}\left(\frac{x}{\alpha}\right)^{c-1}\left[1+\left(\frac{x}{a}\right)^{c}\right]^{-(k+1)} \qquad x>0, \alpha>0, c>0, k>0$$

La figura 4 muestra la forma de esta función para distintos valores de los parámetros.

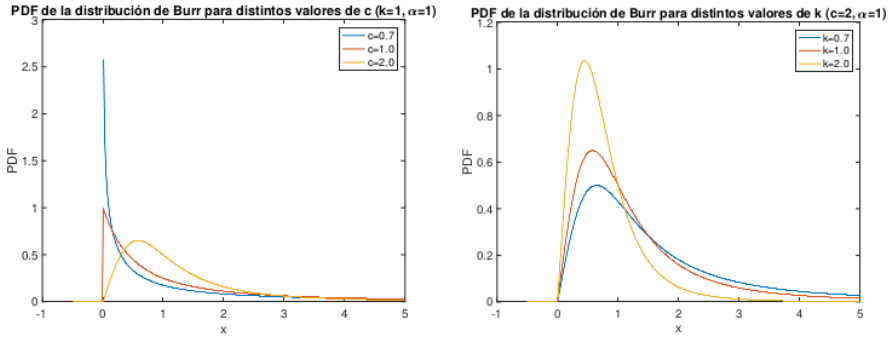


Figura 4: PDF de la distribución Burr con $\alpha = 1$: izquierda: dependencia con c, derecha: dependencia con k

La correspondiente distribución acumulada, viene dada por:

$$F^{Burr}(x) = 1 - \left[1+\left(\frac{x}{\alpha}\right)^{c}\right]^{-k}$$

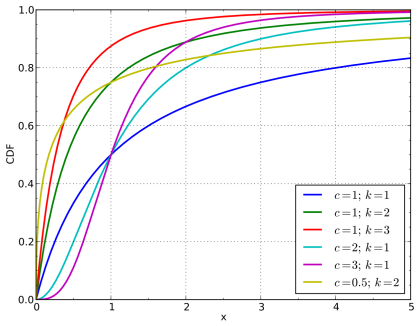


Figura 5: CDF de la distribución Burr con $\alpha = 1$: dependencia con c y k

Su dependencia con los parámetros se muestra en la figura 5[13].

En la figura 6[10], se recoge un ejemplo de ajuste de las pérdidas de datos debido a incendios en Bélgica a una distribución Burr.

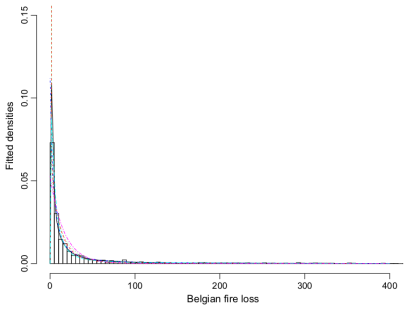


Figura 6: Ajuste de pérdidas de datos debido a incendios en Bélgica a una distribución Burr

Como se aprecia en ella, el ajuste Burr aproxima bien aquellos conjuntos de datos con distribuciones de cola pesada. No obstante, se deberá comprobar la calidad del ajuste. Esto último se analizará en la sección 3.2.3.

2.3 Algoritmos de Clasificación: K-Means

Otro de los motivos de este estudio, ha consistido en caracterizar el tráfico de red mediante mixturas *Burr*. Para conseguir el análisis de mixturas, se ha procedido a realizar la segmentación y clasificación de los datos.

La segmentación de los datos[14] consiste en particionar un conjunto de datos en varios subconjuntos de acuerdo a determinados criterios de decisión. En este caso, el criterio a escoger es el del algoritmo *K-Means*, un algoritmo de aprendizaje no supervisado cuya característica es el desconocimiento *a priori* de los datos de entrada, y que se define así[15]:

Sea $\{a_1, a_2, ...a_n\}$ *un conjunto de puntos. La suma del cuadrado de las distancias del punto* a_i *a cualquier punto* x *es igual a la suma de los cuadrados de las distancias del centroide del punto* a_i *más* n *veces la distancia al cuadrado desde* x *hasta el centroide:*

$$\Phi_{kmeans}(C) = \sum_{j=1}^{k} \sum_{a_i \in C_j} d^2(a_i, C_j)$$

donde C_j es el centroide asociado a a_i.

El algoritmo de Lloyd define la estrategia natural a seguir para realizar la clasificación mediante *K-Means*:

1. Definir el número de clústeres, k, en los que se dividirá el conjunto con k centroides escogidos al azar.
2. Reagrupar los datos en los distintos clústeres en función de la distancia a su centroide.
3. Calcular la distancia media de los datos a cada centroide.
4. Repetir los pasos 2 y 3 hasta que la distancia media sea mínima.

En la figura 7 se muestra un ejemplo del resultado de la aplicación del algoritmo *K-Means* para clasificar un conjunto de datos en dos dimensiones y con tres clústeres[15].

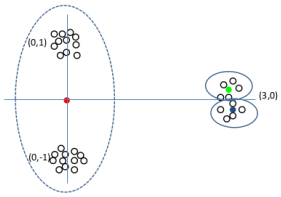


Figura 7: Ejemplo de $K - means$ aplicado a tres clústeres

Pese a ser un algoritmo orientado a la cuantificación vectorial, esto es, para conjuntos de datos de varias variables, se ha comprobado también su utilidad para reagrupar y clasificar datos para funciones de una variable.

3 Diseño y desarrollo

3.1 Introducción

Todo trabajo de investigación requiere de un proceso de planificación que fije los métodos y herramientas tanto de producción como de análisis para poder examinar la hipótesis formulada en el origen de su planteamiento. Por ello, se dedica todo este capítulo para dar constancia de dicho proceso.

En esta parte del trabajo se va a desgranar de manera pormenorizada todos los pasos que se dieron para alcanzar el objetivo de esta investigación. Para ello, se ha decidido dividir este capítulo en dos secciones: una dedicada al diseño, que tratará de la puesta a punto de todo el proceso de preparación para acometer el objetivo en cuestión; y otra al desarrollo, donde se mencionan los contratiempos surgidos y las decisiones tomadas. Sobre este último asunto, no obstante, se volverá a incidir a lo largo del capítulo 4.

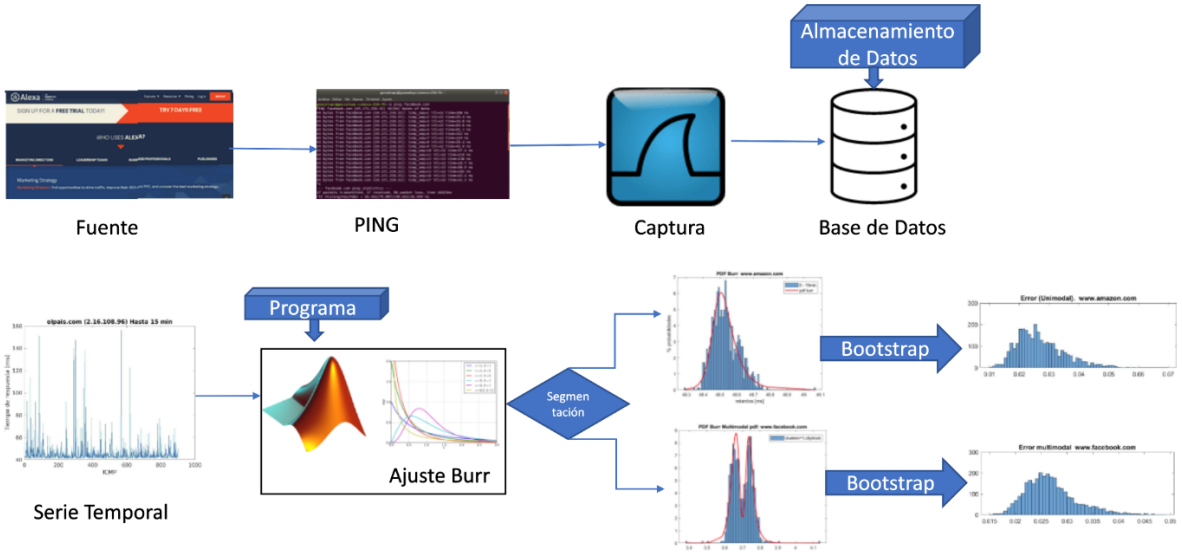


Figura 8: Esquema del diseño del problema

3.2 Diseño del problema

A fin de poder demostrar que la función de distribución Burr permite un buen ajuste de los datos de tráfico de red, este trabajo se ha estructurado en tres tareas consecutivas (ver fig 8). En primer lugar se ha procedido a la captura de datos y su almacenamiento en una base de datos apropiada para su procesamiento posterior. En segundo lugar se han elaborado códigos en MATLAB para procesar los datos capturados y ajustarlos a distribuciones de tipo Burr. Por último, con estos códigos, se ha analizado la calidad de los ajustes mediante el método de *bootstrap*.

Para la captura se ha empleado la herramienta *Wireshark*, o su versión en ventana de comandos para sistemas UNIX, denominada *tshark*, que proporcionan datos de tiempo de tráfico con precisión suficiente

para este estudio (por debajo de los microsegundos).

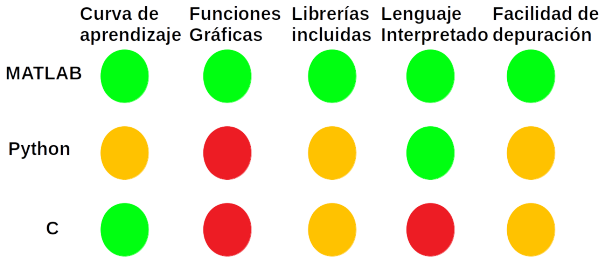


Figura 9: Diagrama RAG comparando `MATLAB`, `Python` y `C`

La elección de MATLAB como herramienta para el procesado de los datos viene justificada por la comodidad que supone el disponer de todas las piezas matemáticas necesarias (en particular las diferentes funciones de ajuste y de tratamiento estadístico) en un único entorno de programación. Asimismo, MATLAB incorpora también funciones de representación gráfica y visualización. A esto se unen las facilidades de MATLAB para la depuración, por tratarse de un lenguaje interpretado. Por otra parte, las desventajas que, desde el punto de vista del tiempo de cálculo, pueda presentar frente a los lenguajes compilados se ven compensadas por el hecho de que las funciones de ajuste están incorporadas a MATLAB, y por tanto son piezas ya compiladas y optimizadas. Tal y como muestra la figura 9.

3.2.1 Captura y almacenamiento de datos.

Una de las características principales de este trabajo ha consistido en la adquisición de los datos y la conformación de una base de datos propia, consistente en ficheros *.txt* sobre las capturas del tráfico de red realizadas. Es decir, no se ha accedido a datos de ningún repositorio o colaborado con el traspaso de datos con ninguna entidad.

El análisis consistió en filtrar las capturas de paquetes ICMP: el protocolo de mensajes de Internet asociado al comando `ping` a un servidor de un dominio web.

Fuente: Alexa.com

Como se muestra en el esquema del Diseño del Problema (fig. 8), lo primero que se hizo fue acceder al portal web alexa.com. Este sitio web, perteneciente a Amazon Web Services, genera un ránking de las páginas web más visitadas a nivel global. Del mismo modo, permite acceder a la visualización de ránkings a nivel nacional. En el caso tratado en este estudio, se usó como fuente de partida el ránking en España accesible en el siguiente dominio:

`http://www.alexa.com/topsites/countries/ES`

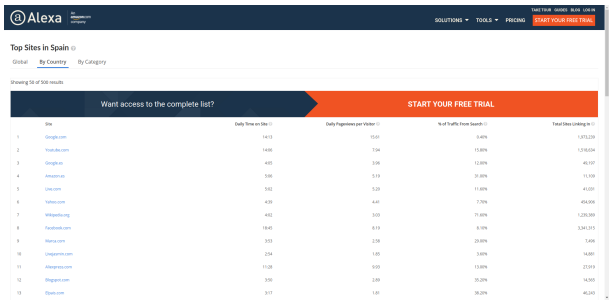


Figura 10: Web alexa.com

El portal web alexa.com se define en su propia página web como un *analizador intuitivo que transforma los datos en conocimiento e información accesible para sus usuarios*[16]. Al cabo de los años, se ha llegado a convertir en el analizador de servicios web más robusto y preciso para cualquier proveedor. El tráfico de red estimado se basa en los datos obtenidos de un panel de tráfico global que tiene la empresa, consistiendo en millones de muestras de usuarios de Internet. Añadiendo a esta información, la propia página web alexa.com afirma que se recopila la información de fuentes directas conformadas por aquellos sitios web que optaron por instalar el script alexa. Por último, se menciona que el ránking de visitas consiste en una medida de cuál es el desempeño del sitio web analizado en los últimos tres meses, usando una metodología que combina el número de visitantes únicos promedio en el día con el número de páginas visitadas. Alexa.com permite una gran variedad de opciones orientadas a la gestión y mejora del desempeño de una página web, aunque en este caso se tratará el ránking de webs más visitadas para realizar el análisis.

A partir de los resultados ofrecidos por este portal, se seleccionaron varios dominios web con alto índice en el ránking en dos fechas distintas y desde dos localizaciones distintas (ver tabla 1). En la primera semana de enero de 2019 se efectuaron las capturas desde una red local doméstica, y entre los días 9 a 15 de marzo de 2019, desde una red de campus universitario.

Campus Universitario	LAN doméstica
elpais.com	elpais.com
wikipedia.org	google.com
live.com	twitter.com
twitter.com	facebook.com
amazon.com	youtube.com
blogspot.com	amazon.com
okdiario.com	blogspot.com
facebook.com	live.com
youtube.com	wikipedia.com

Tabla 1: Dominios consultados

Para la captura de tráfico de red en una red de campus universitario se accedió mediante conexión vía

`ssh`. No obstante, no se ha pretendido inferir los diferentes comportamientos del tráfico de red entre un tipo de red u otro, pero se ponen a disposición de futuros trabajos de investigación los datos obtenidos, por si ellos sirvieren para tal propósito.

PING

Sobre esos dominios se puso en marcha una consecución de *pings* y se procedió a capturar el tiempo de respuesta por parte del servidor que en ese momento estaba atendiendo las peticiones. En la figura 11 se muestra un ejemplo de captura para el caso de facebook.com

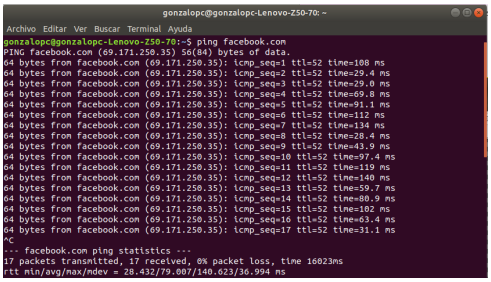


Figura 11: Ejemplo de `ping` en GNU-Linux

El funcionamiento del comando `ping` y su relación con el protocolo de control de mensajes de internet (ICMP) se han expuesto en el Capítulo 2. El estudio del tráfico de red mediante *pings* al *dns* y no al servidor permite observar comportamientos plausibles de que se estén usando varios servidores de forma simultánea, o bien alternándose en función de la congestión del tráfico de red que hubiese en el momento de la captura realizada. En este caso se observarán ciertos comportamientos que dan como resultado histogramas con forma bimodal.

Para la captura de datos desde un terminal accediendo a otro terminal remoto, se barajaron posibilidades como conexión mediante proxy (foxyproxy), comandos en Windows (Putty), etc. Pero se determinó que con hacer `ssh` desde el terminal principal al remoto bastaba.

El comando `ssh`, o *secure shell*, permite el acceso remoto a un servidor o un nodo en general por medio de un canal seguro y con conexión desde una aplicación o línea de comandos. Hay diversos métodos de realizar la conexión `ssh` desde distintos sistemas operativos y se puede realizar tanto dentro de una red LAN como de una red a otra (como fue el caso que se siguió en este trabajo de investigación), habilitando los distintos permisos de usuario sobre los documentos.

Como se acaba de mencionar, `ssh` es un canal seguro, es decir, viene encriptado por el protocolo de seguridad de la capa de transporte (TLS, aunque antes se refería a SSL)[17] frente alternativas inseguras de conexión como pudiera ser Telnet[18].

El comando `ssh` se usó para crear los ficheros a partir de los `ping` y la simultánea captura con tshark, y los ficheros creados se transfirieron desde el nodo remoto al terminal del usuario mediante una copia segura con el comando `scp` basado en el mismo protocolo que `ssh`.

Captura: Wireshark

La captura de pings por parte del terminal de GNU/Linux padecía de una excesiva *granularidad* y poca precisión lo que daba lugar a resultados imperfectos. Para mejorar la precisión de captura de datos, como alternativa al comando `ping` en terminal, se empleó el software Wireshark y de su versión en terminal de

GNU/Linux tshark.

Wireshark, según su propia web, se define como el analizador de protocolos de red más famoso y usado, y permite un conocimiento de lo que ocurre en una red a nivel microscópico. También permite una inspección profunda de cientos de protocolos, así como la captura en streaming, el análisis *offline*, el uso multiplataforma y una gran variedad de opciones.

En la figura 12 se muestra el resultado de una captura del tráfico de red (filtrado sobre los mensajes ICMP tanto enviados como recibidos).

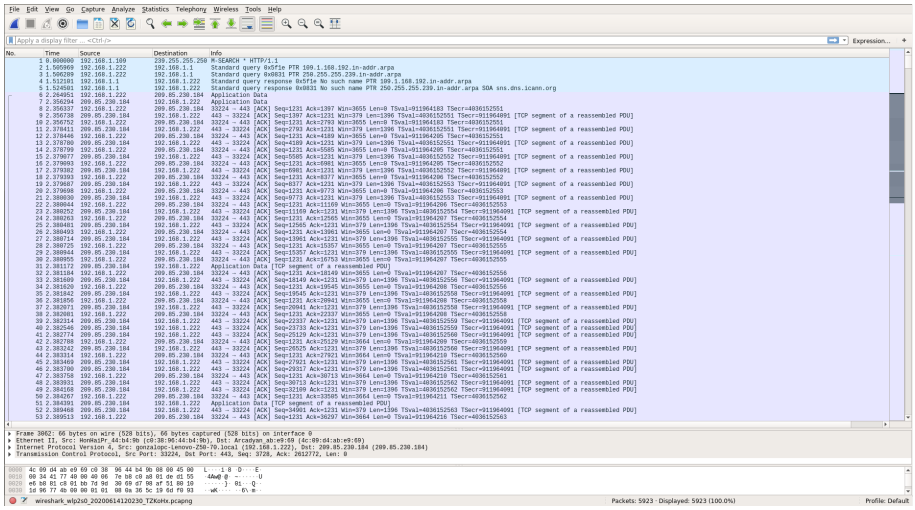


Figura 12: Captura de tráfico de red con wireshark

A diferencia de la limitada precisión del comando `ping` en el terminal, que solo capturaba tres dígitos, Wireshark permite capturar el tráfico de red con distintos niveles de granularidad. Para el caso que tratamos aquí, Wireshark mostraba resultados del orden de los nanosegundos.

Combinando la ejecución simultánea de `ping` en un terminal con la ejecución de *tshark* en otro, se procedió a capturar el tráfico de respuestas ICMP y exportarlo a un fichero de texto mediante el siguiente comando:

```
sudo tshark -i [interfaz] -Y [condiciones y protocolos] -T fields \
-e icmp.resptime > [fichero.txt]
```

A continuación, se muestra un ejemplo de dicho comando para el caso de la captura de amazon.es desde un interfaz wifi:

```
sudo tshark -i wlp2s0 -Y "icmp.type==0 && ip.src_host contains \"amazon.es\"" -T fields \
-e icmp.resptime >amazones_05012019.txt
```

Por otro lado, en el caso de la captura realizada desde un terminal en la red del campus universitario, esto se hizo mediante una conexión `ssh` de la siguiente manera:

```
sudo tshark -i any -Y "icmp.type==0" -T fields -e icmp.resptime >facebook_prueba.txt
```

Una de las características de un sistema distribuido es la transparencia de distribución, que consiste en aparentar que su servidor sea un todo y no un conjunto de servidores. Dicha característica de transparencia se consigue mediante el uso de DNS (Domain Name Service), capaz de asignar IP's de diversos servidores a un mismo dominio.

Para el análisis de tráfico, se han tenido en cuenta los dominios expuestos en la tabla 1 y se ha implementado una comunicación mediante varios pings en torno a las tres o cuatro horas de duración. En la figura 13 se muestra una comparativa entre los datos obtenidos a través del terminal (izquierda) y mediante Wireshark (derecha). En ella se aprecia claramente la diferencia de precisión entre ambos procedimientos tal y como se mencionó anteriormente.

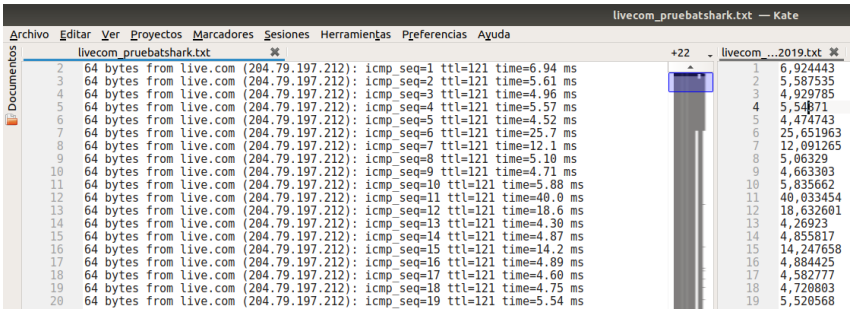


Figura 13: Comparativa de tiempos de respuesta en terminal (izquierda) frente a wireshark (derecha)

Base de Datos:

Tras el proceso de adquisición de los datos, mientras se ejecutaban las operaciones de captura del tráfico de red mediante la interfaz de captura de *tshark*, se fueron generando diversos ficheros de texto (.txt) con los resultados obtenidos tanto de la IP del servidor como los resultados del tiempo de respuesta a cada `ping` ejecutado. Todos estos ficheros de texto se fueron agrupando en una sencilla base de datos (ver figura 14) para poder ser gestionada posteriormente, y cuyo análisis se comentará en el apartado 3.3.

Conviene resaltar que, a diferencia de otros trabajos realizados sobre este problema, aquí se ha trabajado con una Base de Datos propia generada para la ocasión. Esto nos ha permitido analizar otros resultados de interés como los relativos a los tráficos cuya representación requiere el uso de mixturas bimodales.

```
PING amazon.com (205.251.242.103) 56(84) bytes of data.
64 bytes from amazon.com (205.251.242.103): icmp_seq=1 ttl=229 time=130 ms
64 bytes from amazon.com (205.251.242.103): icmp_seq=2 ttl=229 time=130 ms
64 bytes from amazon.com (205.251.242.103): icmp_seq=3 ttl=229 time=128 ms
64 bytes from amazon.com (205.251.242.103): icmp_seq=4 ttl=229 time=129 ms
64 bytes from amazon.com (205.251.242.103): icmp_seq=5 ttl=229 time=129 ms
64 bytes from amazon.com (205.251.242.103): icmp_seq=6 ttl=229 time=136 ms
64 bytes from amazon.com (205.251.242.103): icmp_seq=7 ttl=229 time=127 ms
64 bytes from amazon.com (205.251.242.103): icmp_seq=8 ttl=229 time=128 ms
64 bytes from amazon.com (205.251.242.103): icmp_seq=9 ttl=229 time=128 ms
64 bytes from amazon.com (205.251.242.103): icmp_seq=10 ttl=229 time=128 ms
64 bytes from amazon.com (205.251.242.103): icmp_seq=11 ttl=229 time=133 ms
64 bytes from amazon.com (205.251.242.103): icmp_seq=12 ttl=229 time=130 ms
64 bytes from amazon.com (205.251.242.103): icmp_seq=13 ttl=229 time=130 ms
64 bytes from amazon.com (205.251.242.103): icmp_seq=14 ttl=229 time=135 ms
64 bytes from amazon.com (205.251.242.103): icmp_seq=15 ttl=229 time=129 ms
```

Figura 14: Ejemplo de archivo de la Base de Datos

3.2.2 Programación del problema.

Como se ha expuesto en el apartado anterior, la base de datos generada consiste en el conjunto de series temporales asociadas a cada dominio web. Sin embargo, las series temporales contienen un gran volumen de datos abarcando una gran ventana de tiempo. Para simplificar el tratamiento, se procede a realizar el análisis en ventanas de tiempo en torno a los 15 minutos.

Programa: MATLAB

Todo el procesamiento de los datos se ha realizado con el entorno de programación de MATLAB. Con lenguaje propio, este software matemático está muy bien orientado al uso en el mundo de la Ciencia de Datos y al análisis estadístico, como es el caso que compete a este estudio. Su lenguaje interpretado no requiere de compiladores, y el programa se ejecuta línea a línea, lo que facilita el proceso de depuración.

Frente a otras opciones de lenguajes de programación orientadas al análisis de Ciencia de Datos, como pudiera ser Python, MATLAB posee la ventaja de tener todas las funciones y scripts asociados en la biblioteca principal.

Otra de las características que justifican la elección de MATLAB es la inclusión de un *sandbox* o entorno de pruebas, con diversas Apps accesibles. Entre ellas, una orientada al ajuste de funciones a unos datos datos: *distribution fitter* (Ver figura 15).

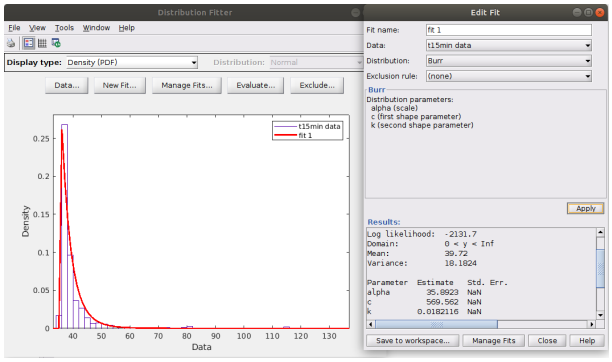


Figura 15: Sandbox Distribution Fitter de Matlab

3.2.3 Ajuste de los datos: distribución Burr.

En el capítulo 2 se ha definido y desarrollado todo el material estudiado con respecto a la distribución Burr. En este punto, simplemente se hará una observación sobre el ajuste a los datos producido por MATLAB y sus elementos característicos: la funcion *fitdist*, el ajuste Burr en MATLAB y las ventajas del tiempo de ejecución frente a alfa-estable.

Función fitdist

Esta función, perteneciente al catálogo o librería de funciones de MATLAB, está basada en el *sandbox/app distribution fitter*. Se ejecuta o no dependiendo de si los datos a los que va a ajustar siguen realmente o no una función de distribución Burr. En caso de que sí se ejecute, la función devuelve un objeto o estructura con tres parámetros asociados: α, c, k, que intervendrán posteriormente en la función *función de densidad de probabilidad* (*pdf, probability density function*).

Función pdf

La función *pdf* devuelve un vector columna con los valores asociados a la función de densidad de probabilidad para los parámetros α, c, k del ajuste Burr asociados a la serie temporal analizada. Esta función será comparada visualmente con el histograma obtenido.

A partir de la *pdf*, se obtiene la *función de probabilidad acumulada* (*cdf, cumulative density function*), que se comparará con la *función de probabilidad acumulada empírica* (*ecdf, empirical cumulative distribution function*) y servirá para evaluar la la calidad del ajuste mediante el método de *bootstrap*.

Ajuste Burr vs. alfa-estable

En la tabla 2 se muestra una comparativa de tiempos de generación de series temporales con el ajuste Burr y con el ajuste alfa-estable. En cada iteración se generan 1000 simulaciones.

n	LAN		Campus Universitario	
iteración	Burr	alfa-estable	Burr	alfa-estable
1	1.5488	5.3316	2.0391	1.5265
2	1.7557	10.956	1.529	5.1173
3	2.1809	12.542	1.4757	9.4528
4	2.0518	13.233	1.4029	10.56
5	1.6165	14.349	1.4042	11.029
6	1.456	15.392	2.0859	15.864
7	1.9713	13.696	3.0044	14.232
8	1.5097	11.426	1.8803	13.631
9	1.7765	11.88	1.6037	1.0602
10	1.627	11.89	1.667	15.235
11	1.4277	11.307	2.4158	13.151
12	2.0314	13.705	3.3481	13.707

Tabla 2: Tiempos de ejecución (en segundos) para twitter (LAN) y facebook (Campus Universitario)

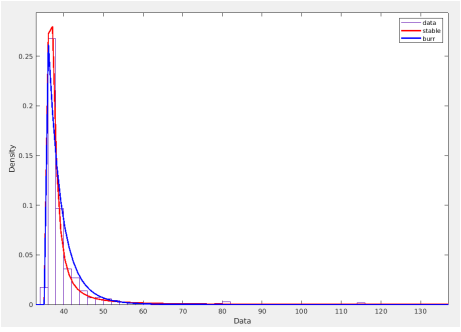


Figura 16: PDF Burr (azul) y α-estable (rojo)

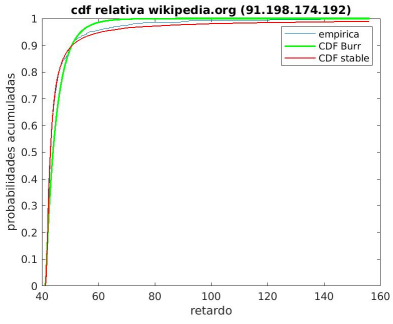


Figura 17: CDF Burr (verde) y α-estable (rojo)

Por otra parte, en las figuras 16 y 17 se comparan las *pdf* y *cdf* de ambas distribuciones.

Observando la tabla 2 y las figuras 16 y 17, se llega a la conclusión de que, desde el punto de vista del tiempo de cálculo, el uso del ajuste Burr es ventajoso con respecto al alfa-estable para distribuciones de

cola pesada, y la calidad de ambos ajustes es similar, lo que corrobora la hipótesis del planteamiento de este estudio.

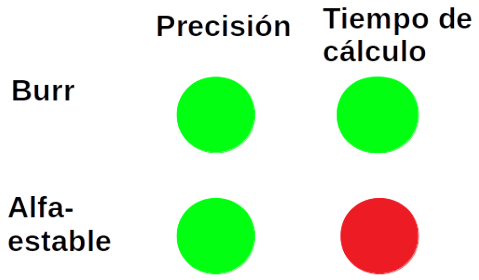


Figura 18: Diagrama RAG comparando las ventajas de la distribución `Burr` vs la `alfa-estable`

La figura 18 muestra mediante un diagrama RAG la elección de la distribución `Burr` en detrimento de `alfa-estable` del terminal.

Análisis de las ventanas de tiempo (15 minutos)

Para procesar los datos y realizar el análisis, se procede a particionar el conjunto de valores de la serie temporal asociada al sitio web en ventanas de intervalos de aproximadamente 15 minutos, obteniéndose así subconjuntos de 900 datos con los que se realiza el análisis para cada intervalo de tiempo.

Sobre estas series temporales de 15 minutos, se procede a agrupar los datos en histogramas con 100 barras (ver figura 19), después de comprobar que los resultados no difieren significativamente de los que se obtienen con otras reglas establecidas, como pudiera ser la regla Freadman-Diaconis o la raíz cuadrada del número de elementos asociados al conjunto.

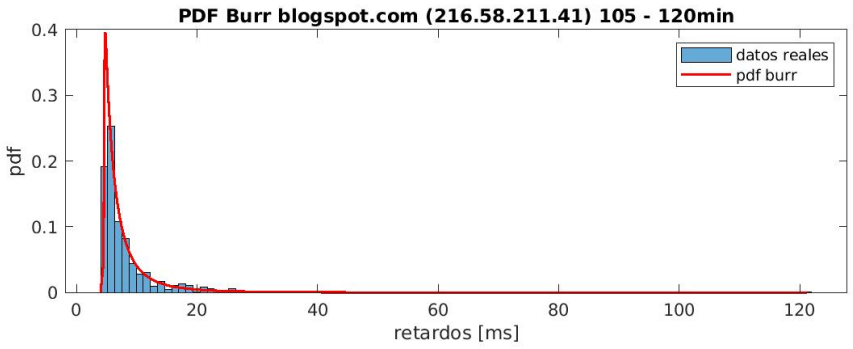


Figura 19: Ajuste de PDF de Burr (rojo)

Análisis de mixturas

Hay algunos casos en los que el resultado visual obtenido de la función de densidad de probabilidad permite inferir que se puede tratar el problema desde una perspectiva de análisis de mixturas, como se aprecia en la figura 20. Estos casos tal vez pueden atribuirse a la presencia de varios servidores emisores de paquetes de datos y posibles variantes de la ruta de tráfico de red.

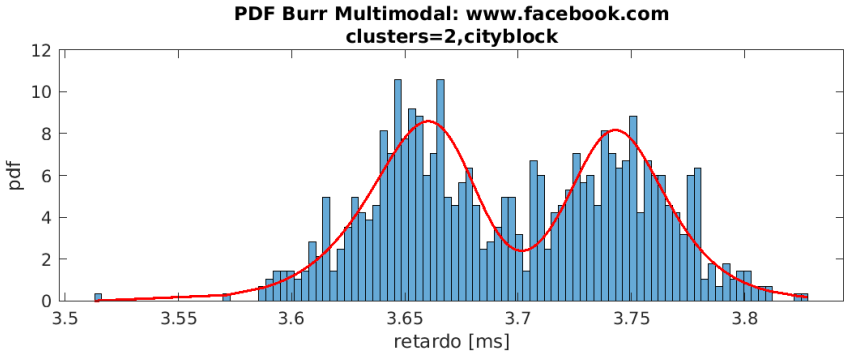


Figura 20: Ejemplo de mixtura Burr simétrica

Sin embargo, no todas las mixturas son tan perceptibles o claras como la bimodal representada en la figura 20. En muchos casos, es la propia máquina la que detecta un posible cluster imperceptible a los ojos humanos, o que puede ser confundido con alguna pequeña componente de ruido estadístico.

Segmentación: K-Means

En el capítulo 2, se ha hecho un breve repaso al algoritmo K-Means y su campo de aplicación. Aquí, se va a inscribir dicho algoritmo en el análisis pertinente al presente estudio.

Lo primero que se debe mencionar sobre el flujo del programa en relación a la segmentación es que esta no tiene por qué producirse. En otras palabras, es decisión de la propia función de ajuste fitdist si los datos son susceptibles de ser modelados como una mixtura Burr. Para el caso de que sea así, se han generado dos funciones encadenadas: *clusterBimodal* y *calculoKmeans*. Estas dos funciones fueron creadas *ad hoc* para el análisis desde el punto de vista de segmentación y reagrupamiento de los datos en clústeres bajo ciertos criterios de distancias calculadas respecto a un centro estimado. Estas funciones permiten la posibilidad de crear tantos centroides para cada cluster como se deseen y desarrollar el análisis para cada caso.

En la mayoría de los casos de estudio, no se produce una estimación de mixtura o, en el mejor de los casos, se puede segmentar como una bimodal asimétrica, con un clúster principal donde está el grueso de los datos concentrado en valores de retardo menores y un clúster secundario que se observa como un leve repunte de los datos, tal y como se muestra en la figura 21.

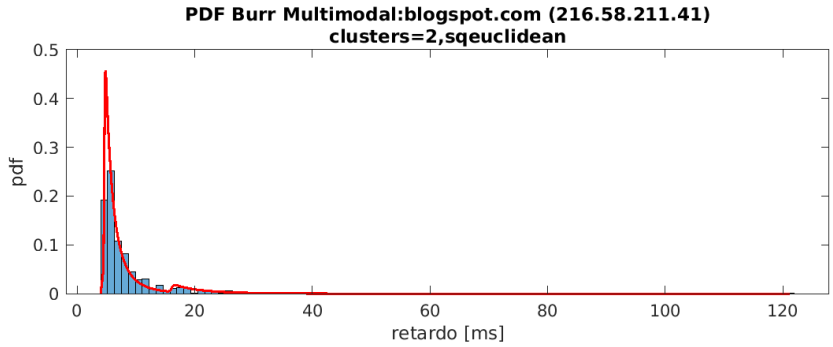


Figura 21: Ejemplo de mixtura Burr asimétrica

También se han encontrado casos de ajustes para tres clústeres o incluso cuatro, aunque estos son menos habituales y no los incluimos aquí. Sin embargo, uno de los resultados más significativos es la obtención de mixturas bimodales con un número de datos asociados a cada centroide bastante equilibrado. Este último caso corresponde a capturas de datos en la red de campus universitario, aunque se deberá ser prudente a la hora de llegar a alguna conclusión con respecto a este resultado.

La función *clusterBimodal*, requiere como datos de entrada la serie temporal, el número de clústeres que se van a calcular y otros parámetros útiles para la función *calculoKmeans*. *clusterBimodal* realiza, mediante la función *kmeans* incluida en el catálogo de MATLAB, la segmentación de los datos y el cálculo del centroide asociado a cada clúster siguiendo diversos criterios de cálculo de la distancia. Sobre este último punto, se debe aclarar que en este estudio tratamos con funciones de una variable, a diferencia de otros trabajos, como por ejemplo el de la rerferencia [7], que trata sobre funciones de dos variables. Esto se traduce en que la definición de distancia queda restringida a unas pocas posibilidades, en comparación con los casos de análisis multivariante. En este punto, conviene recordar que la finalidad de este estudio era el ajuste de la función de distribución Burr al tráfico de red tomando como única variable el retardo de llegadas de mensajes de respuesta ICMP.

La definición matemática de distancia entre dos puntos, A y B, en un espacio cualquiera viene dada por las condiciones:

$$d_{AB} \geq 0; \qquad d_{AB} = d_{BA}; \qquad d_{AB} + d_{BC} \geq d_{AC}$$

En nuestro caso consideraremos dos definiciones de distancia entre centroides: la asociada a la *geometría del taxista* también llamada *Manhattan Metrics* (*cityblock* en MATLAB) y la *sqeuclídea*. La distancia *Manhattan Metrics* viene dada por

$$d_{AB}^{MM} = \sum_{i=1}^{n} |A_i - B_i|$$

y la distancia *sqeuclídea* (*'sqeuclidean'* en MATLAB) se define como:

$$d_{AB}^{sqe} = \sum_{i=1}^{n} (A_i - B_i)^2$$

En ambos casos A y B son vectores de n dimensiones.

Análisis de la mixtura Burr

Para los casos en los que se haya podido segmentar los datos para un número de clústeres mayor o igual que dos, se procede a realizar un análisis similar al que se hizo sin el criterio ni la suposición de la segmentación. Cada clúster de datos es analizado aparte, calculando los parámetros (α, c, k) de la distribución Burr, para obtener a continuación sus PDF y CDF.

De todos los clústeres asociados con los datos, se realiza una ponderación normalizada para dar más peso a aquellos que cuenten con un mayor número de datos.

Con estos elementos se procede a realizar la mixtura Burr de todos esos clústeres ajustando cada elemento a una distribución Burr. Para ello se sigue el criterio de suma de PDF tal y como se indica a continuación:

$$PDF_{multi} = \frac{1}{n_{tot}} \sum_{i=1}^{N} n_i \; f_i(x) \; dx$$

donde n_i es el número de elementos asociado a cada clúster, $n_{tot} = \sum_{i=1}^{N} n_i$, f_i es la PDF del clúster i-ésimo, y N es el número de clústeres.

Una vez calculada la PDF es ajustada al histograma de los datos. Y la CDF multimodal es comparada con la CDF unimodal y la ECDF de los datos originales.

3.2.4 Calidad del ajuste: el método bootstrap.

Una vez realizado todo el proceso del ajuste de los datos a una distribución Burr, se debe evaluar la calidad de este ajuste y estimar cuál es el error más probable con este método, para confirmar o descartar si el método de ajuste es válido para el análisis. Si la distribución Burr proporciona un error similar al de alfa-estable, teniendo en cuenta el coste de cálculo claramente inferior, significará que Burr es una elección útil, eficiente y fiel para el ajuste del tráfico de red como sustituto de alfa-estable.

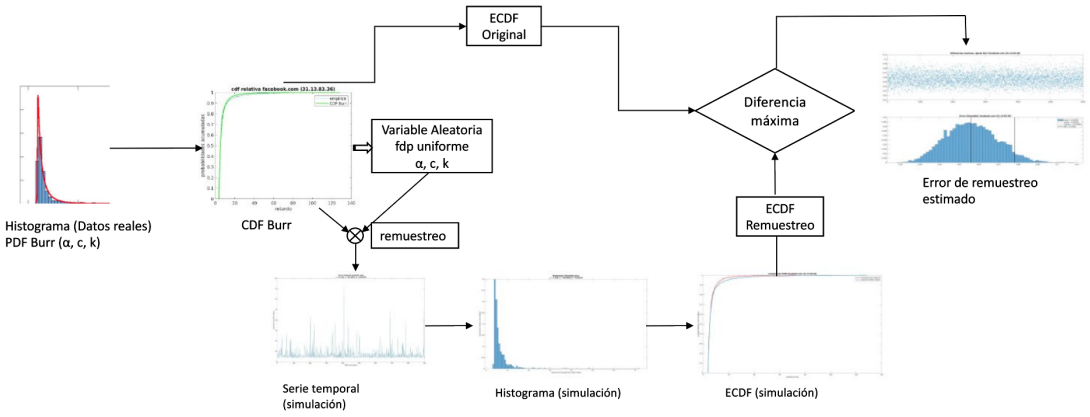


Figura 22: Esquema del método *bootstrap* de estimación del error de ajuste

La figura 22 ilustra la forma de proceder en el método *bootstrap*. Como se aprecia en ella, el proceso busca romper la dependencia entre la función *ecdf* y su correspondiente *cdf* teórica mediante el remuestreo

de esta última y la generación de múltiples series temporales. Estas series se generan siguiendo el criterio de una distribución Burr con los parámetros (α, c, k) obtenidos del ajuste de los datos empíricos.

El primer paso consiste en la obtención del histograma que agrupa los datos reales de la serie temporal y el ajuste de esos datos a la función de distribución Burr teórica. A partir de esa PDF, se calcula la correspondiente CDF.

En segundo lugar, se procede a realizar el remuestreo de la señal. Para ello se genera una variable aleatoria siguiendo una distribución uniforme en el intervalo $[0, 1]$ para los valores de y. Se generan tantos valores en ese intervalo como datos tiene la serie temporal, y sobre este vector de valores aleatorios obtenidos se calculan los valores de x correspondientes a la inversa de la CDF de Burr, tal y como se muestra en la figura 23. Para este propósito se emplea la función *icdf* incluida dentro de MATLAB:

```
x_bootstrap_burr = icdf('burr',y_bootstrap,alpha_burr,c_burr,k_burr);
```

donde `x_bootstrap_burr` es el resultado obtenido de realizar la operación del cálculo de la cdf inversa a partir del vector de elementos aleatorios con PDF uniforme, `y_bootstrap`. Los restantes datos, `alpha_burr`, `c_burr` y `k_burr`, son los parámetros (α, c, k) de la distribución Burr para los datos reales obtenidos según el método que se desarrolló en el punto 3.2.1.

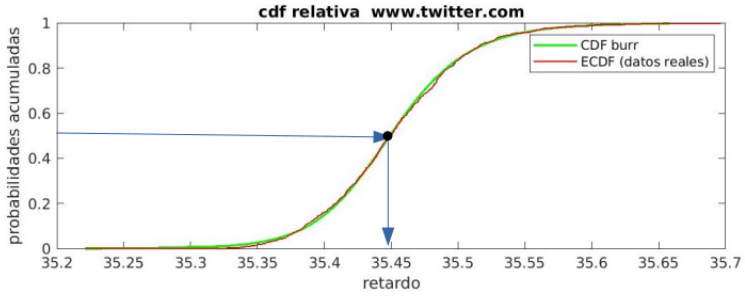


Figura 23: Esquema del método *bootstrap* de estimación del error de ajuste

Con los valores nuevos remuestreados de x se genera una serie temporal que puede ser analizada como cualquier otra serie temporal, es decir, recalculando su PDF, hallando su histograma, calculando su ECDF y efectuando las demás operaciones que se realizan con los datos medidos.

En la figura 24 muestra un ejemplo de comparación entre la ECDF de las medidas reales y la del ajuste siguiendo el método de remuestreo.

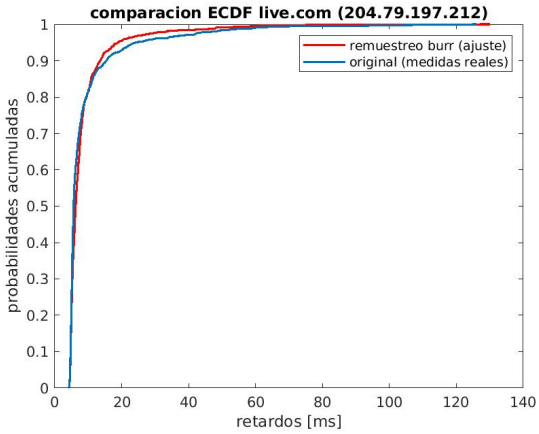


Figura 24: ECDF simulada (rojo) vs. ECDF real (azul)

En tercer lugar, se debe calcular la distancia máxima entre el valor en el eje y de la función ECDF de los valores medidos y los de la función ECDF de los valores remuestreados. Ese resultado de la distancia máxima será considerado como el *error de remuestreo*.

Todo este procedimiento se realiza para un número significativamente grande de iteraciones. De esta forma se conforma un algoritmo de cálculo del error asociado al ajuste a una distribución en concreto. Es importante resaltar que el número de iteraciones, N, debe ser un valor elevado (se hacen pruebas con valores $N = 5000, 6000, 8000, ...$) para conseguir valores determinantes sobre el ajuste. El procedimiento culmina con la obtención de una función de distribución del error asociado al remuestreo tal y como se observa en la figura 25. En esta figura, el eje x indica los valores del error de remuestreo, a partir de los cuales se pueden extraer datos estadísticos: error medio, mediana, varianza, etc.

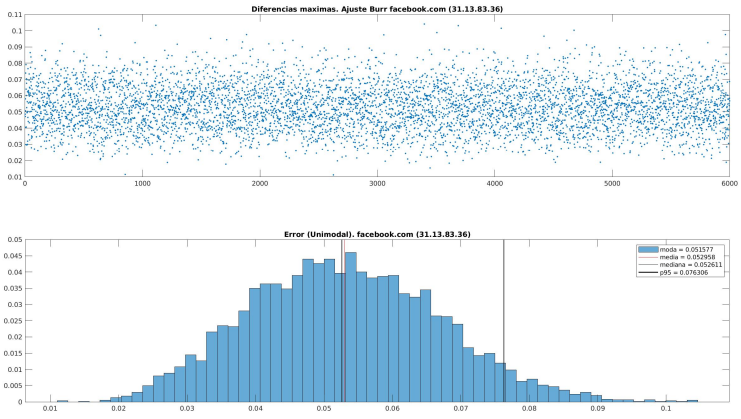


Figura 25: Error estimado mediante el método *bootstrap*

3.3 Desarrollo.

El diseño del problema fue encontrándose con diversos contratiempos y la necesidad de reconfigurar la estrategia implementada a la hora de alcanzar los objetivos a medida que avanzaba su realización. Esta sección se dedica a ir exponiendo las diversas adversidades, la decisión que se tomó en cada caso y el porqué de esa decisión.

3.3.1 Problemas encontrados durante el almacenamiento de datos

En primer lugar, la decisión de crear una base de datos con capturas de series temporales propias acarreaba la necesidad de tener que automatizar el proceso de captura de los datos.

Se podría haber acelerado el proceso de automatización de captura si se hubiese implementado previamente algún método asociado al Web Scrapping que ofrecen lenguajes como `python` para capturar datos de páginas html. Este método habría permitido, mediante la creación de un script de `python` la extracción de los dominios de las webs más visitadas según el ránking *alexa.com* y su posterior tratamiento desde ese fichero *script shell*.

Por otro lado, como se mencionó con anterioridad en la subsección 3.2.1, el comando `ping` del terminal se encontraba con la limitación de solamente proporcionar valores de 3 dígitos, independientemente del orden de magnitud y de la precisión. Por ello se decidió usar Wireshark, un software que permite resolver el problema de granularidad y baja resolución del `ping` de GNU/Linux y calibrar la precisión hasta donde se desee.

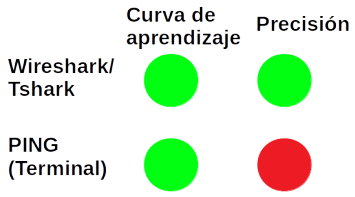


Figura 26: Diagrama RAG comparando `Wireshark` vs `ping`

En la figura 26 se muestra mediante un diagrama RAG la decisión de elegir `Wireshark` en vez del uso del `ping` del terminal.

3.3.2 Decisiones sobre Segmentación

El resultado del procesamiento obtenido del análisis los datos hacía intuir la posibilidad de atacar el problema desde el punto de vista de las mixturas. Por este motivo se estudió si se podían segmentar los datos y ajustar cada clúster a una distribución Burr para después obtener una PDF Bimodal.

En este caso, se eligió el algoritmo de clusterización *K-Means* en detrimento de *K-Medians* por dar el primero resultados más razonables en la definición de los centroides. Por último, la elección del método de cálculo de distancias: *cityblock* o *sqeuclidean* resultó irrelevante, como ya se explicó en la subsección 3.2.2, ya que, al realizarse un análisis de una función de una variable, la restricciones impuestas por la definición de una distancia matemática quedaban acotadas a una dimensión y, en este caso, más allá de la elección aleatoria del centroide, los resultados obtenidos con ambas distancias tendían a ser iguales.

La distancia cuadrática euclídea frente a la distancia euclídea solo se justifica por razones de eficiencia de cálculo, ya que la segunda implica tomar una raíz cuadrada sin añadir nada a la calidad de los resultados.

3.3.3 Decisiones sobre el método bootstrap

El proceso de remuestreo según el método bootstrap ya se explicó detalladamente en la subsección 3.2.4 Se consideró la posibilidad de remuestrar la señal con un número significativamente mayor de valores que los incluidos en la serie temporal de los datos reales medidos. Esta posibilidad se desechó porque conducía a comportamientos extraños, como es la aparición de distribuciones de los errores con más de un máximo. Esto se ilustra en la figura 27. Finalmente, para un intervalo de 15 minutos, que equivale a 900 muestras, se optó por remuestrear la señal original con una nueva de 1000 muestras.

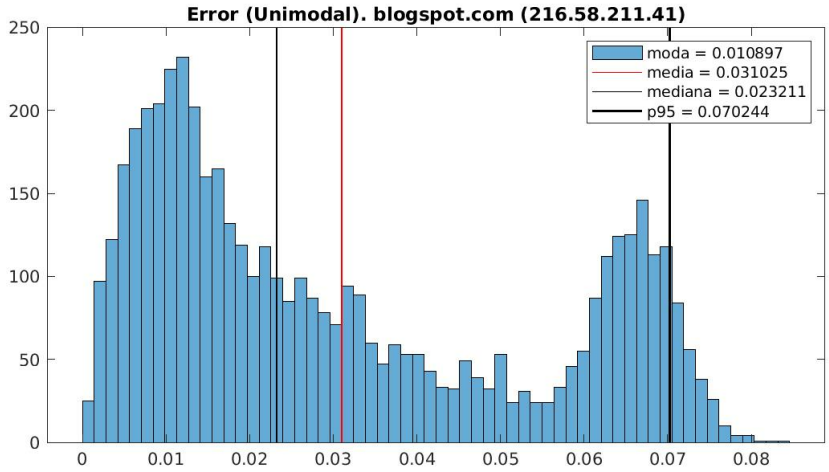


Figura 27: Error estimado para una simulación con 2500 muestras

Por último, esta creación de la serie temporal simulada necesitaba ser comparada con la serie temporal original y ejecutada un gran número de iteraciones. Se hizo para valores entre 5.000 y 10.000 iteraciones (siempre teniendo en cuenta el tiempo y esfuerzo de cálculo que conlleva un aumento del número de iteraciones) siguiendo los postulados del teorema central del límite y la ley de los grandes números.

4 Resultados y validación

4.1 Introducción.

En este capítulo se aplican las técnicas descritas en el capítulo 3 a dos problemas, un ajuste unimodal que también puede aproximarse como bimodal asimétrico y otro claramente bimodal, con el fin de mostrar la validez del diseño metodológico propuesto en este trabajo.

Este capítulo se divide en tres secciones en las que se abordan la adquisición y almacenamiento de los datos, el ajuste a distribuciones unimodales y el ajuste a distribuciones multimodales. Esta última se

divide a su vez en dos subsecciones dedicadas la las distribuciones bimodales simétricas y asimétricas, respectivamente.

4.2 Datos reales.

En el capítulo 3, dedicado al diseño y el desarrollo del estudio, se explicó el procedimiento y la metodología de adquisición de datos de tráfico de Internet mediante el filtrado de paquetes ICMP (el protocolo de mensajes de control de internet, asociado al comando `ping` ofrecido desde distintos sistemas operativos, en este caso, GNU/Linux).

La adquisición y almacenamiento de datos desde un terminal situado en una red de área local doméstica, permite obtener resultados interesantes sobre el análisis del tráfico de red particularizados a este ámbito. De esta manera, la elaboración propia de una base de datos partiendo desde cero (*from scratch*) permite obtener series temporales que, una vez caracterizadas, dan como resultado distribuciones claramente bimodales, tal y como se verá en la sección 4.4. Esta es una de las contribuciones originales del presente trabajo con respecto a otros estudios relacionados con el mismo campo.

Aquí nos proponemos no solo mostrar el resultado de los datos y el ajuste a una distribución Burr de tres parámetros, sino ilustrar la bondad del propio ajuste. Para ello, se estimará el error asociado a cada ajuste mediante la técnica de remuestreo bootstrap expuesta en la sección 3.2.4.

Recapitulando lo anteriormente expuesto, recordemos que en el método bootstrap se deben generar un número relativamente grande de series temporales simuladas a partir de la CDF teórica que ajustada los datos reales capturados. La diferencia máxima entre la ECDF de esta nueva serie temporal simulada y la ECDF de la serie temporal original se usa como una medida del error (ver figura 22 en la sección 3.2.4). Si los datos reales permiten un buen ajuste a una PDF de tipo Burr, la nueva señal simulada se asemejará mucho a la original y la diferencia entre ambas, que será pequeña, cuantificará el error de remuestreo.

Sin embargo, con la simulación de una serie temporal a partir de la CDF teórica no basta. Es necesaria la realización del cálculo para un número elevado de iteraciones. Cuanto más elevado mejor, aunque con la contrapartida de un mayor coste computacional. Un número razonable de iteraciones estaría comprendido en una horquilla entre 5.000 y 10.000, lo que permitiría aplicar la Ley de los Grandes Números y su aproximación al Teorema Central del Límite. Este teorema dice que *si S_n es la suma de n variables aleatorias independientes y de varianza no nula pero finita, entonces la función de distribución de Sn "se aproxima bien" a una distribución normal* [19].

Se observa que la distribución del error estimado aproxima a una distribución gaussiana. En algunos casos, a una función gaussiana asimétrica con asimetría positiva tal y como se expone en la figura 28.

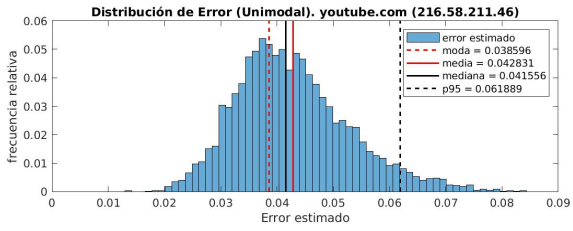


Figura 28: Distribiución de errores con asimetría postiva

En este algoritmo es donde más se hace notar la mayor velocidad de cálculo de la distribución Burr con tres parámetros frente a la de alfa-estable, mucho más lenta e ineficiente y que habitualmente requiere el cálculo en paralelo, tal y como indica el trabajo de Bizumuremyi Herrero[20]. Los resultados del tiempo de ejecución comparativos entre ambas distribuciones se mostraron en la tabla 2 de la sección 3.2.3.

Con todo esto, en las siguientes secciones exponemos los ejemplos más ilustrativos de los resultados del análisis.

4.3 Ajuste Unimodal.

En esta sección se trata el análisis de series temporales que se ajustan a una distribución unimodal de tipo Burr. Los elementos característicos de la distribución Burr ya fueron comentados en el Capítulo 2, aquí mostramos algunos casos representativos de este tipo de ajuste.

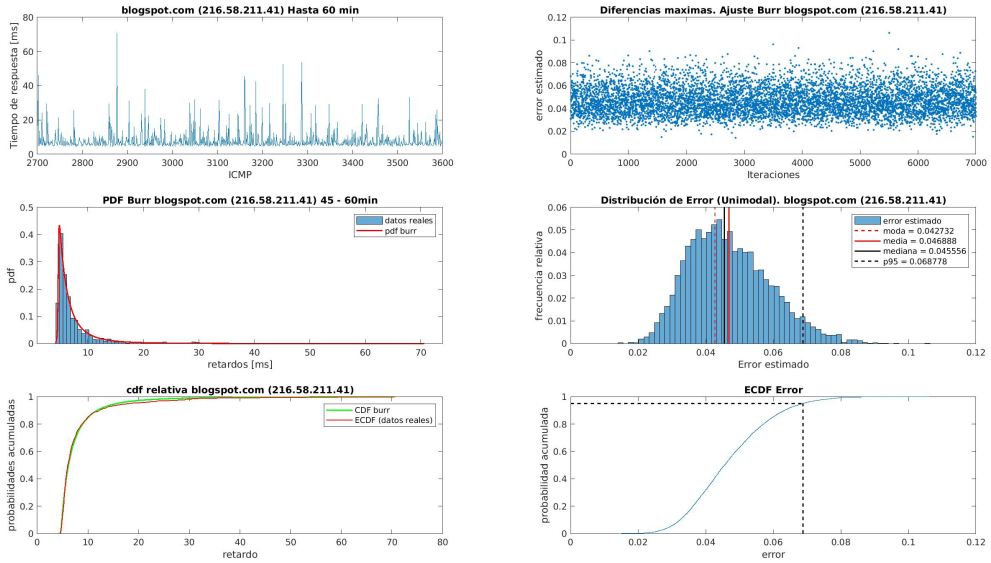


Figura 29: Ajuste Burr a tráfico de internet: Red LAN, Blogspot 45-60 min

La figura 29 muestra los resultados de la captura de tráfico de red de blogspot.com para un intervalo de tiempo de 15 minutos (entre el minuto 45 y el 60 de captura). En la imagen superior izquierda, puede observarse la serie temporal, cuyos datos se han reagrupado en un histograma que se muestra en la imagen inmediatamente inferior. Este histograma ha sido normalizado debidamente para poder efectuar el ajuste a una función Burr mediante MATLAB. La función *fitdist* incluida en las librerías de MATLAB se ha usado para estimar los parámetros (α, c, k) correspondientes a la función de distribución Burr ajustada.

Como se puede apreciar en la figura, se trata de una función asimétrica o de cola pesada para valores de retardo correspondientes a los percentiles más altos de la distribución. En algunos casos, se pueden detectar repuntes en esas colas pesadas, tal y como se verá en la sección 4.4.

La tercera gráfica de la columna izquierda corresponde a la CDF de la distribución Burr comparada

con la ECDF de los datos reales. Como se explicó en la sección 3.2.4, la bondad del ajuste estimado por MATLAB se analiza mediante el algoritmo de *bootstrap* y se cuantifica en términos de la diferencia máxima respecto a la serie temporal de los datos empíricos.

Las diferencias máximas vienen representadas por puntos en la gráfica superior derecha, habiendo tantos puntos como iteraciones se llevan a cabo. A partir de estos valores, se genera la *distribución del error* que se observa en la gráfica central derecha. El uso de un número suficientemente grande de muestras para tener una distribución representativa busca aproximarse a las condiciones requeridas para que se puedan aplicar el teorema central del límite o la ley de los grandes números. Como se aprecia en este caso, la distribución del error estimado sigue una distribución gaussiana con algo de asimetría positiva (la media se sitúa en valores mayores que la mediana y ésta respecto de la moda). Por otro lado, el valor representativo del percentil 95 se da para estimaciones de error del 7%. Es decir, en el 95% de los casos, la estimación del error del ajuste será menor del 7%. Aunque el valor promedio esté por debajo del 5%. Hay ajustes de tráfico de red que han mostrado incluso valores inferiores a los datos del ajuste en este caso, lo que sustenta la fiabilidad del ajuste de la distribución Burr a un flujo de tráfico en Internet tal y como pretendemos mostrar en este estudio. Por último, en la imagen inferior derecha, se muestra la ECDF de la distribución del error correspondiente a la gráfica anterior.

El comportamiento observado en la figura 29 se observa también en otros casos de captura desde una red LAN doméstica como se muestra en las figuras 30 y 31.

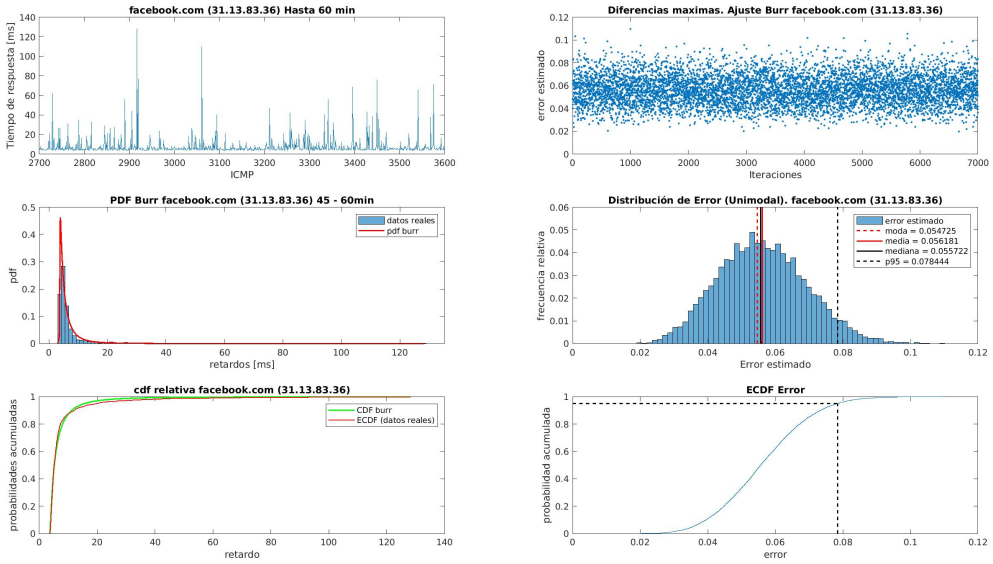


Figura 30: Ajuste Burr a tráfico de internet: Red LAN, Facebook 45-60 min

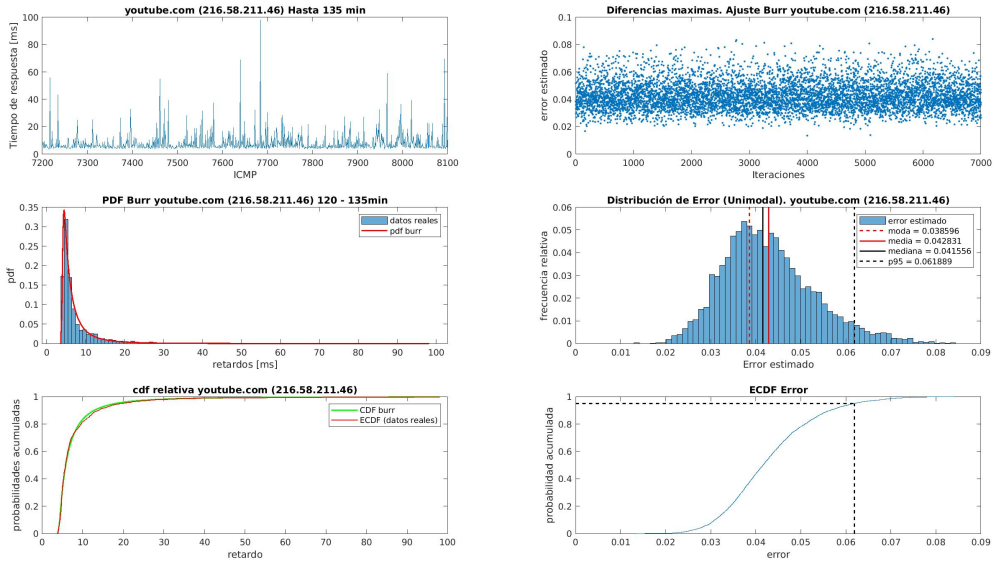


Figura 31: Ajuste Burr a tráfico de internet: Red LAN, Youtube 120-135 min

Por contraste, en los casos de capturas desde la red de campus universitario, las distribuciones se aproximan más a una gaussiana como se aprecia en las figuras 32 y 33. Esto parece sugerir que el tipo de comportamiento es característico del entorno de despliegue de red en el que se ha realizado la adquisición de los datos, si bien este estudio no se dedica a indagar en el porqué de esto.

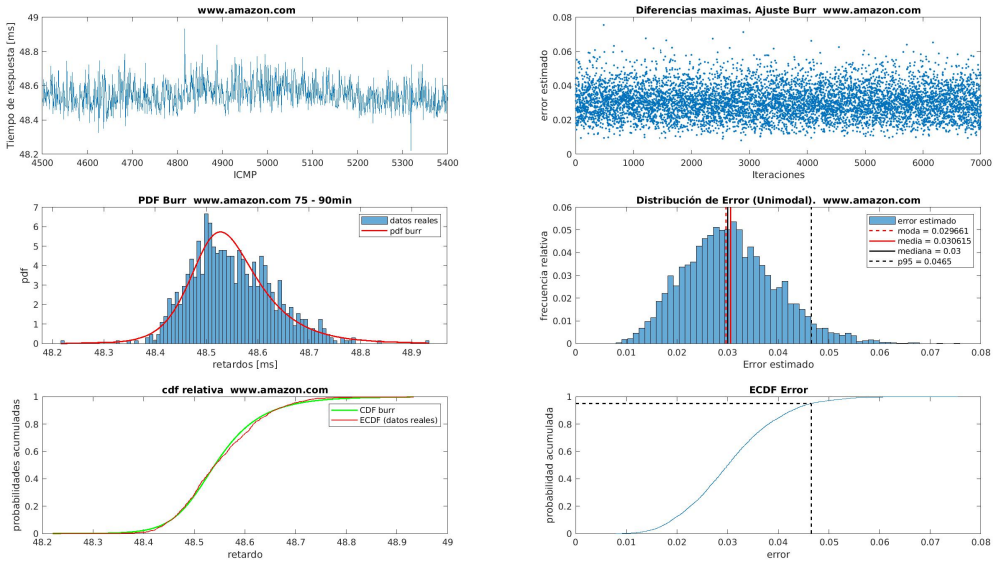


Figura 32: Ajuste Burr a tráfico de internet: Red Campus Universitario, Amazon 75-90 min

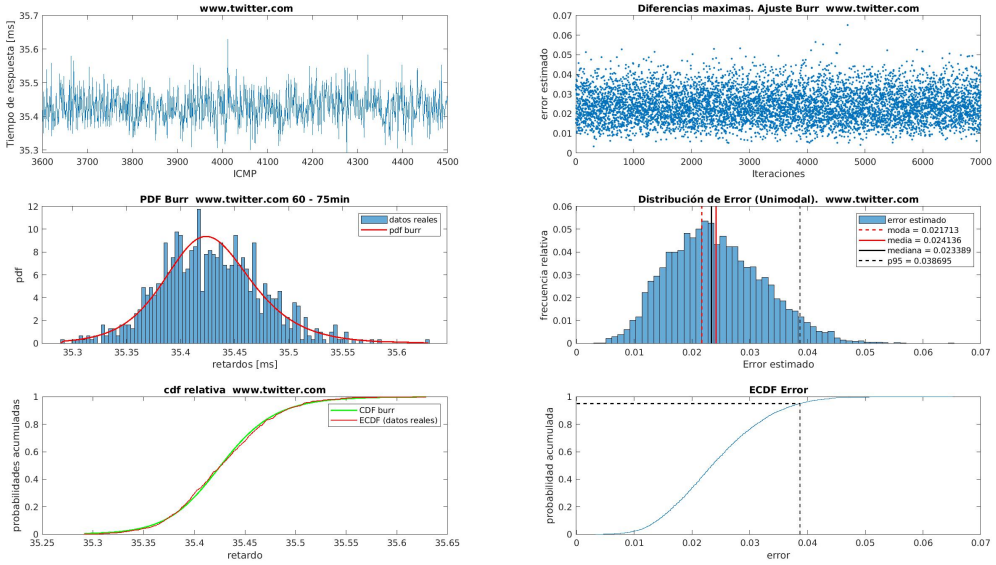


Figura 33: Ajuste Burr a tráfico de internet: Red Campus Universitario, Twitter 60-75 min

Como se ve en las figuras 32 y 33, los datos adquiridos en la red del campus universitario arrojan un valor medio del error entre el 2,5% y el 3%, y una probabilidad del 95% de que el error esté en torno al 4% (o incluso por debajo de este valor). Conviene resaltar además que, pese a tener la apariencia de una

distribución gaussiana, todo el ajuste se ha realizado usando una distribución Burr.

Como ya se ha mencionado, la mayor o menor asimetría de la distribución de error no permite extraer conclusiones determinantes en relación con el entorno de captura de los datos, es decir, si se trata de la red doméstica o de la red del campus universitario. No obstante, lo que sí se aprecia en ambos casos son los valores pequeños del error (incluso inferiores al 3%), por lo que, al menos para distribuciones unimodales, la función Burr puede ser clasificada como una buena función de ajuste para los datos de tráfico de Internet capturados con independencia del entorno de captura. A esto cabe añadir las ventajas que da su eficiencia de cálculo con respecto a la distribución alfa-estable.

4.4 Ajuste Multimodal.

Del mismo modo en que se analizaron los resultados obtenidos para los casos en que el tráfico de red se podía modelar con una función de distribución Burr, se puede proceder para aquellos conjuntos de datos que han ofrecido como resultado mixturas, bien sean bimodales claramente simétricas, asimétricas, o incluso multimodales.

Esta sección se divide en dos subsecciones: en la primera estudian los ajustes a mixturas bimodales simétricas y, en la segunda, el de bimodales asimétricas. Para ello, se empieza siempre por realizar el ajuste unimodal, para seguir después con el de la bimodal, y proceder después a la comparación entre ambos ajustes.

4.4.1 Bimodal simétrica

En primer lugar, la figura 34 muestra como quedaría ajustada la serie temporal caracterizada por el histograma sobre los datos reales a una distribución Burr sin la asunción de mixturas o la segmentación de los datos para reagruparlos sobre los clústeres correspondientes. Como elementos a destacar de este primer ajuste están la CDF correspondiente en la tercera gráfica de la columna izquierda y el resultado del error medio estimado: por encima del 10%, con una probabilidad del 95% de tener un error por debajo del 12,5%. Sin embargo, tal y como se expuso en el capítulo 3, el estudio y todo el programa se orientó a demostrar que, no solo el ajuste Burr es bueno y eficiente, sino que se puede modelar el tráfico de red a partir de mixturas Burr.

Para ello, el programa se preparó para posibilitar la segmentación y reagrupamiento de los datos (ver la subsección 3.3.2) siguiendo el algoritmo K-Means (Capítulo 2), con una implementación de *Machine Learning* habitual en problemas de segmentación y clasificación de los datos.

Antes de proceder al análisis de los resultados, recordaremos que, para una clasificación de datos en una función de una variable, da igual el método de cálculo de la distancia de los datos a su centroide asociado, salvo en lo que se refiere a la eficiencia del procedimiento. Por este motivo, en el caso de ofrecer el ajuste de la mixtura, la distancia de cálculo podrá ser la *cityblock* (Geometría del taxista) o la *sqeuclidean* (euclídea cuadrática). Por contra, no considerarmeos la euclídea normalizada, que es menos eficiente en el cálculo de la distancia y no aporta ninguna novedad en los resultados de clasificación.

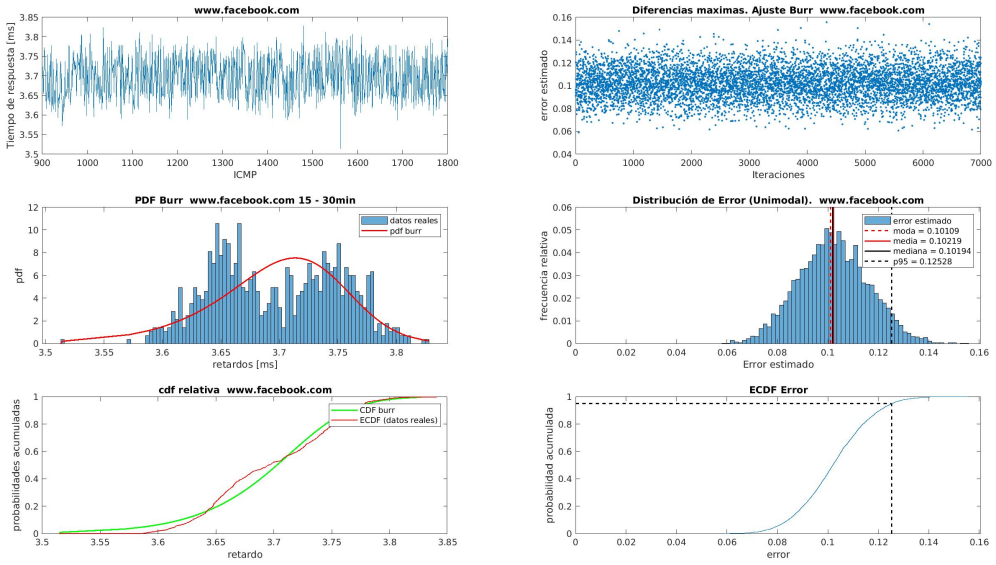


Figura 34: Ajuste unimodal Burr a tráfico de internet: Red Campus Universitario, Facebook 15-30 min

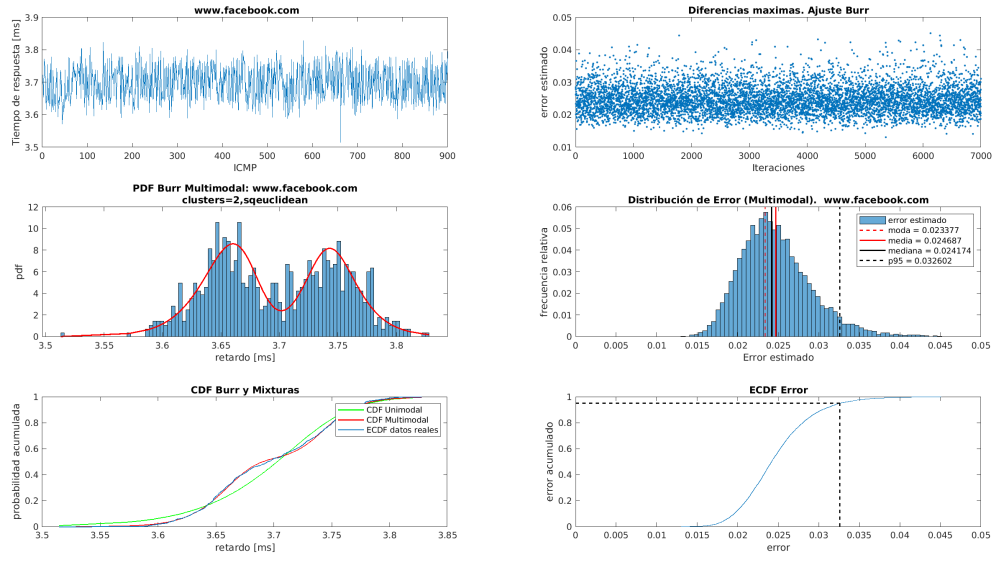


Figura 35: Ajuste mixtura Burr a tráfico de internet: Red Campus Universitario, Facebook sqeuclidean 15-30 min

En la figura 35 se muestra el resultado de aplicar la técnica del algoritmo K-Means para 2 clusters. Una vez calculados los centroides y reagrupado los datos, se ha realizado el ajuste para una mixtura Burr

y se ha analizado la bondad de este ajuste. Como detalle interesante, en la gráfica correspondiente a la comparación entre la CDF del ajuste unimodal, la CDF correspondiente al ajuste multimodal y la ECDF de los datos reales, se observa una evidente mejoría cuando el ajuste se modela como una distribución bimodal.

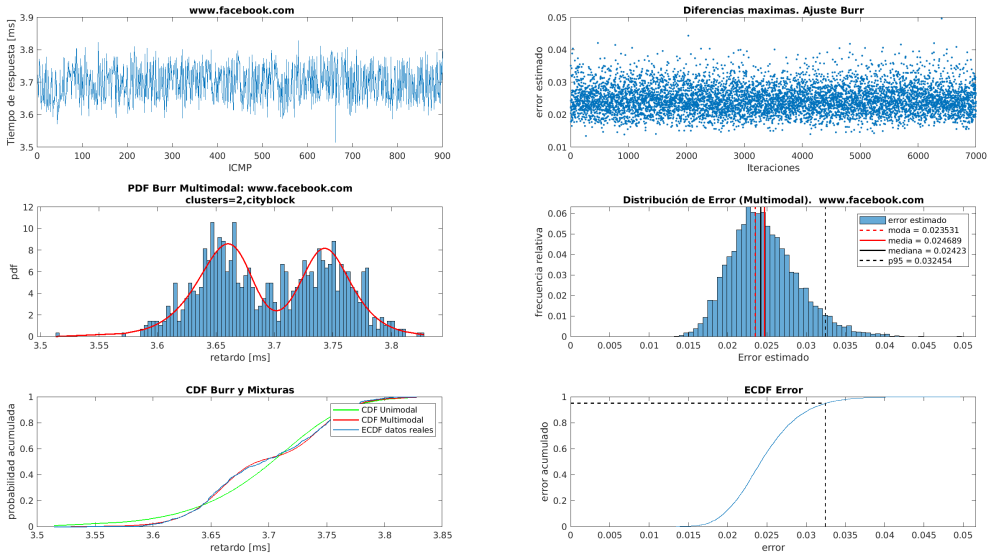


Figura 36: Ajuste mixtura Burr a tráfico de internet: Red Campus Universitario, Facebook cityblock 15-30 min

Además, si en el ajuste unimodal el error medio estimado se situaba en torno al 10%, ahora se obtiene un error de ajuste del 2,5%. Los resultados de la mixtura se ofrecen para un cálculo de la distancia tipo *sqeuclidean*. Los resultados para *cityblock* se ofrecen en la figura 36 y los resultados también arrojan un error estimado promedio del 2,5%. En definitiva, la decisión de recalcular el ajuste de los datos con una función Burr unimodal a una de mixturas Burr puede reducir el error estimado hasta un cuarto.

En la figura 37, se muestran los datos correspondientes al dominio facebook.com para distancia *sqeuclidean*, con un error estimado del 2,5%, y en la figura 38, los del dominio youtube.com para el intervalo de tiempo de captura de 165 a 180 minutos con la distancia *cityblock*, con un error del 2,3%. En todos los casos mostrados, la probabilidad de obtener un error menor del 3,3% es del 95% (percentil 95).

Como observación adicional, hacemos notar la aparición estas mixturas bimodales claramente simétricas, que se han dado en capturas de tráfico de Internet en la red de campus universitario. Aunque este estudio no aborda el porqué de este fenómeno.

Sin embargo, no todas las bimodales quedan tan claras como las mostradas en las figuras [35–38]. Es también habitual encontrarse con mixturas de bimodales asimétricas tal y como se explica en la sección 4.4.2.

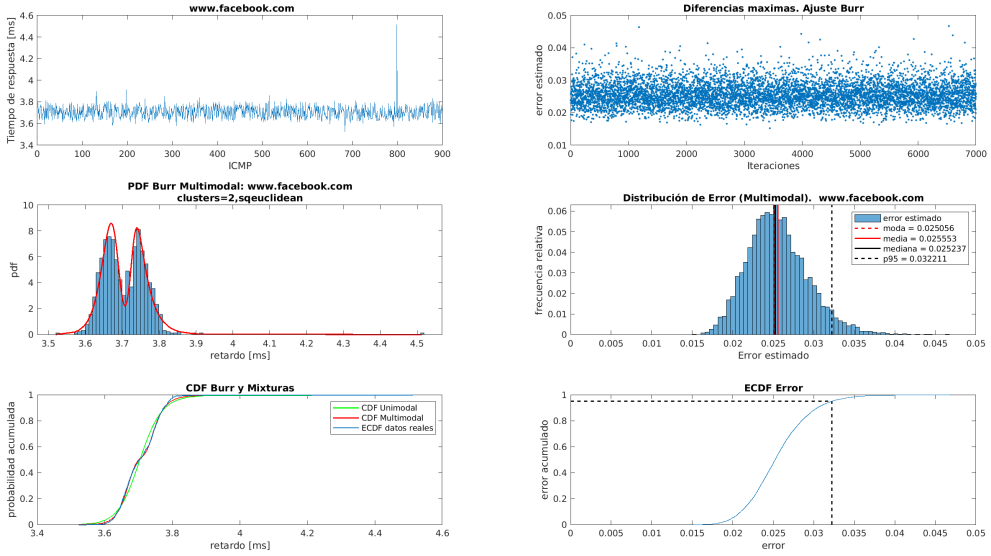


Figura 37: Ajuste mixtura Burr a tráfico de internet: Red Campus Universitario, Facebook sqeuclidean 165-180 min

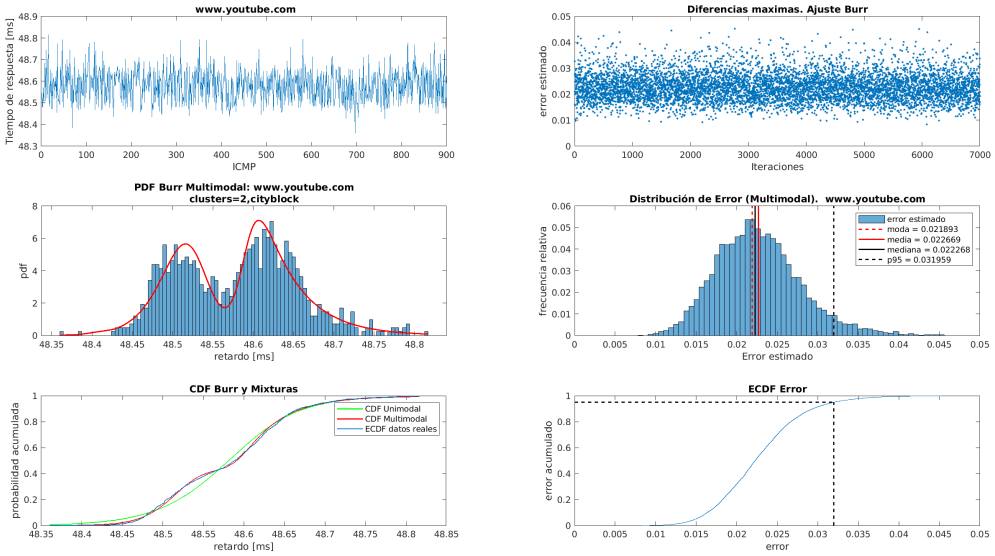


Figura 38: Ajuste mixtura Burr a tráfico de internet: Red Campus Universitario, Youtube cityblock 255-270 min

4.4.2 Bimodal asimétrica

Los casos de bimodales asimétricas se dan, sobre todo, en capturas de datos en entornos de redes domésticas, debido al comportamiento de cola pesada, tal y como se muestran en los histogramas de la serie temporal y en la PDF del ajuste de los mismos. Es en esta cola pesada en la que se suele encontrar el clúster menor, frente al mayor que se observa donde se encuentra el grueso de los datos.

En estos casos, este reajuste no siempre ofrece una mejora en el error estimado, por lo que es menos fiable que el ajuste bimodal mostrado en la sección anterior.

Sin embargo, hay veces en que sí se observa una mejora. Así, en la captura del dominio facebook.com desde la red LAN, la figura 39 muestra un error estimado para ajuste unimodal de un 8,5%, mientras que con el ajuste mediante mixturas Burr asimétricas, el error desciende a un 5,5%, tal como se aprecia la figura 40).

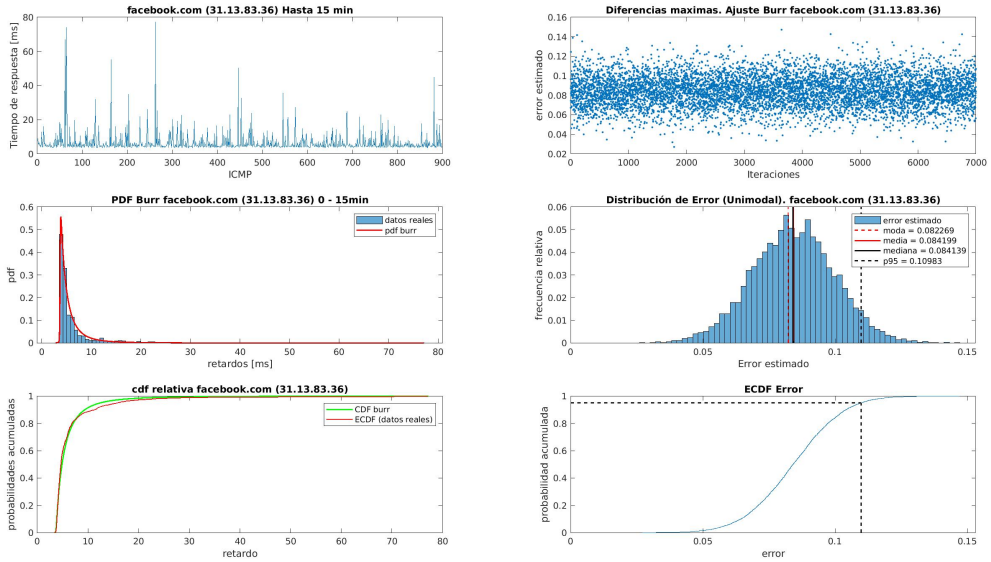


Figura 39: Ajuste Burr unimodal a tráfico de internet: Red LAN, Facebook cityblock 0-15 min

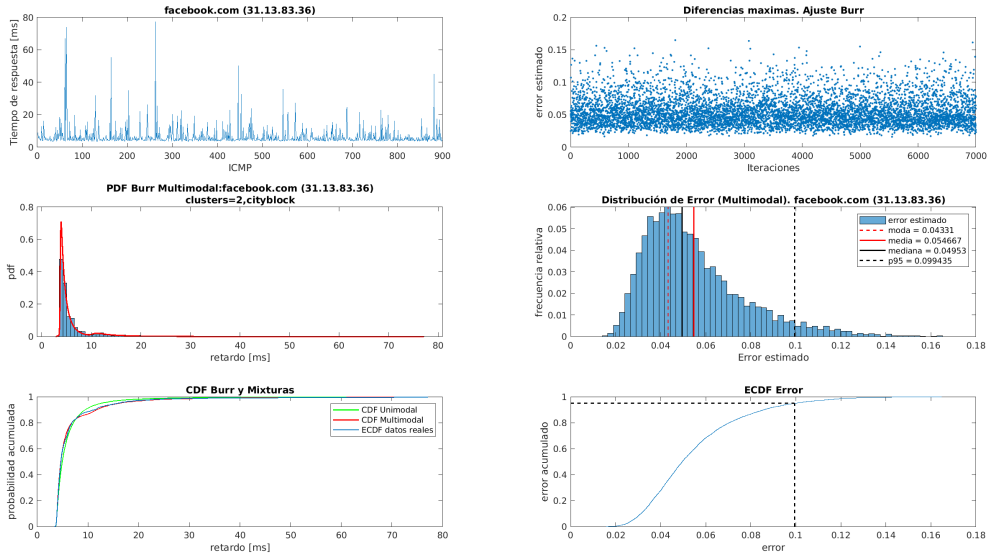


Figura 40: Ajuste mixtura Burr a tráfico de internet: Red LAN, Facebook cityblock 0-15 min

En el ajuste bimodal mediante la mixtura, el máximo del primer clúster se sitúa aproximadamente en los 5 ms de media de retardo por llegada de paquete ICMP de respuesta. Por su parte, el segundo clúster, que no es tan fácilmente perceptible como el primero, se encuentra con un máximo con unos valores de retardo por encima de 12 ms. Atendiendo a lo expuesto por Bizumuremyi Herrero[20]:

La explicación más lógica es dada por el comportamiento del tráfico red. Generalmente existe más de un ruta para llegar a un destino, la ruta escogida por defecto para la invocación y respuesta del comando PING suele ser la más rápida, pero en caso de congestión, pueden escogerse otras rutas, comportamiento que explicaría perfectamente lo sucedido.

Alternativamente, se podría pensar que los servidores a los que se les está demandando la información sean otros (con otra IP real, pero que en la apariencia de los servidores asociados al dominio de una red empresarial de este tipo, se nos ofrezca como un todo). Esto último está ligado con la noción de transparencia de un Sistema Distribuido.

Por último, se observa que la distribución del error para ajustes multimodales suele seguir la forma de una función asimétrica con asimetría positiva, frente a la del ajuste unimodal, que suele aproximarse más a una distribución normal.

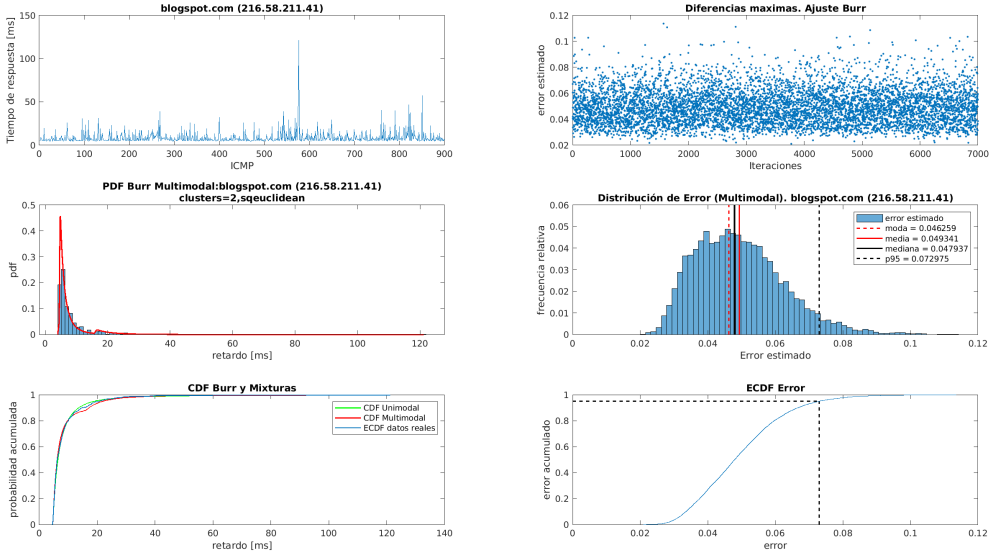


Figura 41: Ajuste mixtura Burr a tráfico de internet: Red LAN, Blogspot sqeuclidean 105-120 min

5 Conclusiones

5.1 Conclusiones.

- A lo largo de este trabajo se ha podido demostrar que el tráfico de red puede ser modelado como una función de distribución Burr de tres parámetros, con buena calidad de ajuste del tráfico de red comprobada.
- También se ha demostrado empíricamente que su cálculo requiere de mucho menos tiempo que el cálculo de una distribución alfa-estable.
- Partiendo de datos de captura propios, obtenidos en este trabajo desde la web alexa.com y extraídos mediante herramientas de análisis de paquetes de red como es el caso de Wireshark, se ha generado un *script* de MATLAB para proceder al estudio y análisis de los mismos. Se ha comparado visualmente el resultado de los ajustes de los datos tanto a la distribución alfa-estable como a la distribución Burr.
- Se ha realizado un test detallado sobre la calidad del ajuste a la distribución Burr siguiendo el método de remuestreo de *bootstrap*.
- Se han comprobado ciertos patrones de comportamiento de la red que se asemejaban a la formación de funciones multimodales.
- Se han estudiado y analizado estos patrones dividiendo el problema en el análisis de subproblemas mediante el algoritmo de clasificación *K-Means* para la posterior síntesis mediante el estudio de mixturas. En este caso, también se ha realizado un test de la calidad del ajuste.
- Se ha intentado contrastar una mejora en la calidad del ajuste para el análisis multimodal frente al análisis unimodal. Sin embargo, no siempre se ha dado el caso de afirmar esta mejora.
- En los casos de captura de tráfico de red desde la red del campus universitario se han llegado a obtener distribuciones claramente bimodales simétricas que pueden resultar muy interesantes para el estudio posterior. En estos casos sí se ha visto una notable mejora mediante el análisis de mixturas frente al unimodal.
- En los casos de captura de tráfico desde la red LAN doméstica se ha observado un comportamiento tal que la caracterización del mismo mediante mixturas no implica una mejora con respecto a la caracterización unimodal.

5.2 Contribuciones.

Como se acaba de mencionar en la sección 5.1, se ha estudiado el análisis mediante mixturas Burr para ajustes multimodales. Pese a que en este trabajo se ha orientado, en este sentido, simplemente al análisis de bimodales simétricas o asimétricas, también se puede caracterizar el tráfico siguiendo funciones multimodales para tres o más clústeres de datos. Por otro lado, se ha obtenido una serie de capturas de tráfico de Internet real que se puede reaprovechar para estudios futuros en relación con este campo de investigación.

Además, se ha escrito una serie de programas para MATLAB en los que se sistematizan los ajustes a distribuciones Burr unimodales y multimodales. Los códigos correspondioentes se recogen en la sección de Anexos.

5.3 Trabajo futuro.

Los distintos patrones de comportamiento observados entre el tráfico capturado en una red doméstica y una red de un campus universitario podrían ser investigados en próximos trabajos que se dediquen a desarrollar este tema para profundizar en el conocimiento del comportamiento de las redes. Sin embargo, este trabajo va encarecidamente orientado para aquellos trabajos futuros orientados a la prevención de ciberataques y de congestión de tráfico, así como a orientar los mismos hacia la automatización de procesos.

6 Bibliografía

References

[1] Una disputa entre empresas ralentiza internet, 2020. https://elpais.com/tecnologia/2013/03/27/actualidad/1364405367_942632.html.

[2] Ataque de denegación de servicio, 2020. https://es.wikipedia.org/wiki/Ataque_de_denegaci%C3%B3n_de_servicio.

[3] Ping (networking utility), 2020. https://en.wikipedia.org/wiki/Ping_(networking_utility).

[4] Internet control message protocol, 2020. https://tools.ietf.org/html/rfc792.

[5] Commandes réseau sous windows et linux, 2020. https://web.archive.org/web/20190518121033/https://blog.pandorafms.org/fr/commandes-reseau/.

[6] Ejemplo de envío y recepción de un ping, 2020. https://es.wikipedia.org/wiki/Protocolo_de_control_de_mensajes_de_Internet#/media/Archivo:ICMPv1.PNG.

[7] Daniel Perdices Burrero. Aplicación de técnicas estadísticas a registros de red para el análisis de tráfico y explotación de datos. Master's thesis, Universidad Autónoma de Madrid, 2018.

[8] Stable distribution, 2020. https://en.wikipedia.org/wiki/Stable_distribution.

[9] Burr Type XII Distribution, 2020. https://es.mathworks.com/help/stats/burr-type-xii-distribution.html.

[10] Shaiful Anuar Abu Bakar, Nor Hamzah, Mastoureh Maghsoudi, and S. Nadarajah. Modeling loss data using composite models. *Insurance: Mathematics and Economics*, 61, 09 2014.

[11] Richard Onyino Simwa, Martin Mutwiri Kithinji, and Joseph Anthony McOtteku Otieno. Application of Burr XII Mixture Distributions to Model Unemployment Duration in Pricing Unemployment Insurance Assuming USA Data. *International Journal of Statistical Distributions and Applications*, 2(3):27–34, 2016.

[12] Pandu R. Tadikamalla. A look at the Burr and related distributions. *International Statistical Review*, 48(3):337–344, 1980.

[13] Cumulative Burr distribution, 2020. https://en.wikipedia.org/wiki/Burr_distribution#/media/File:Burr_cdf.svg.

[14] Alberto Ruiz Santos. Segmentación de tráfico en Internet mediante la clasificación de parámetros estadísticos. Master's thesis, Universidad Autónoma de Madrid, 2019.

[15] Avrim Blum, John Hopcroft, and Ravindran Kannan. *Foundations of Data Science*. 2018. https://www.cs.cornell.edu/jeh/book.pdf.

[16] Alexa: about us, 2020. https://www.alexa.com/about.

[17] The transport layer security (tls) protocol, 2020. https://tools.ietf.org/html/rfc5246.

[18] Teletype network (telnet), 2020. https://es.wikipedia.org/wiki/Telnet.

[19] Teorema del límite central, 2020. https://es.wikipedia.org/wiki/Teorema_del_l%C3%ADmite_central.

[20] Bizumuremyi Herrero. Análisis de tráfico en internet utilizando distribuciones alfa estables en procesadores paralelos. Master's thesis, Universidad Autónoma de Madrid, 2017.

Anexos: programas de ajuste en MATLAB y datos de tráfico

Los códigos correspondientes a análisis de series temporales incluyendo las funciones para ajuste de clústeres unimodales y multimodales se encuentran publicados en el siguiente repositorio:
https://github.com/GonzaloLopezS/Burr_distribution_internet_traffic

Los datos correspondientes a las series temporales capturados entre la primera semana de enero de 2019 y los días 9 a 15 de marzo de 2019 están publicados y disponibles en el siguiente repositorio:
https://github.com/GonzaloLopezS/InternetTrafficData